Host-Adaptive Watermarking based on a Modified Spread Spectrum Technique

A thesis submitted to

The College of Engineering, University of Basrah

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Engineering

BY

Ali Essam Hameed Haddad

B.Sc., University of Basrah (2003)

January 2006

To the memory of my father

CERTIFICATION

I certify that the thesis titled "***Host-Adaptive Watermarking based on a Modified Spread Spectrum Technique***" submitted by ***Ali Essam Hameed Haddad*** has been prepared under my supervision at the ***University of Basrah*** as a partial fulfillment of the requirements for the degree of ***Master of Science in Computer Engineering***.

Signature: -

Supervisor: - Dr. Majid A. Alwan.

Date: -

In view of the available recommendations, I forward this thesis for debate by the examining committee.

Signature: -

Name: - Dr. Majid A. Alwan.

(Head of Computer Engineering Dept.)

Date: -

EXAMINING COMMITTEE'S REPORT

We certify that we the examining committee, have read this thesis, examined the student in its content and that in our opinion it is adequate as a thesis for the degree of ***Master of Science in Computer Engineering***.

Signature: -
Name: - Dr. Jassim M. Abdul-Jabbar.
(Chairman)
Date: -

Signature: -
Name: - Sabah S. Al-sheraidah.
(Member)
Date: -

Signature: -
Name: - Dr. Ramzy S. Ali.
(Member)
Date: -

Signature: -
Name: - Dr. Majid A. Alwan.
(Supervisor)
Date: -

APPROVED FOR THE UNIVERSITY

Signature: -
Name: - Prof. Dr. Saleh E. Najim.
(Dean of the College of Engineering)
Date: -

Acknowledgements

First, of course, I would like to thank my supervisor ***Dr. Majid A. Alwan*** for his guidance, wise advices and useful suggestions throughout the course of this work.

I am also grateful to all my teachers at the Department of Computer Engineering, Engineering College, University of Basrah for their dedication in teaching throughout my undergraduate and postgraduate courses.

I would like to acknowledge ***Prof. Dr. Saleh E. Najim*** the dean of Engineering College and ***Dr. Majid A. Alwan*** the head of Computer Engineering Department, too, for their encouragements and wise management.

Thanks to all my friends and colleagues, especially ***Intessar T. Al-Eadany***, for their encouragements and for many useful discussions.

And finally, I would like to express my gratitude to my family for their patience and unlimited support throughout the course of this work.

Abstract

As digital multimedia get more widely used and also more easily replicated, manipulated and distributed, their copyright protection issues become more challenging. So far, these challenges have mainly been met by the use of encryption algorithms to restrict unauthorized accesses to the intelligible multimedia contents. Encryption algorithms, however, fail to protect these contents as soon as they are decrypted. So, what is needed here is a piece of information that is permanently embedded within the multimedia content, which can be used later to prove ownership of that work. Such a piece of information is well-known today as a *watermark*.

In this thesis, a modified spread spectrum watermarking system is proposed, that can be viewed as a *fined* (*more granular*) version of the *rigid* traditional spread spectrum systems. Here, the traditional power spreading mechanism is *segmented* into a number of *smaller* and *redundant* spreading mechanisms, with the reduction in the processing gain resulting from the latter reduction in the spreading dimensionality being compensated for through the employment of the *integrate-and-dump* technique (well-known in pulse code modulation systems) to take advantage of the available redundancy.

The proposed watermarking system is also extended with a new host-adaptation scheme that aims at enhancing the perceptual quality of the system outcome by *uniformly redistributing the perceptual distortion* -induced by the embedded watermark signal- through the underlying host signal using some adaptation mechanism that does not induce any distortion -by itself- into the embedded watermark, hence, eliminating the need for any adaptation distortion equalization processing prior to watermark decoding.

List of Symbols*

Symbol	Description
$A^N(y^N \mid x^N)$	Attack channel.
B_d, B_q	The bandwidths of the data signal and the spreading sequence, respectively.
C	Channel capacity, the dimensionality of host blocks.
$c_N(m, k^N)$	Coding function.
D	Dimensionality of watermarking (spreading) segments.
D_1, D_2	Distortion constraints on embedding and attack induced distortions, respectively.
$d_N(x^N, y^N)$	Perceptual distortion measure.
d(t)	Data signal.
$\hat{d}(t)$	Estimated data signal.
E^N	Overall channel induced error vector.
F	Number of orthonormal spreading sequences used.
$f_N(s^N, k^N, m)$	Overall embedding (encoding) function.
$f'_N(s^N, u^N)$	Embedding function (code multiplexing section only).
G_p	Processing gain.
$h_N(v^N, s^N)$	Host-adaptation function.
J^N	Additive noise vector.
j(t)	Jamming signal.
K^N	Side information vector.
L	Cardinality (size) of message set.
M	The message to be embedded.
N	Total dimensionality of watermarking vectors (arrays).
P_d, P_j	Total power of data and jamming signals, respectively.

* Refer to Appendix A for the meanings of the uppercase, lowercase, bold, script, subscripted and superscripted letters. Also see Appendix A for the notation used here to express the information content and probability distributions of the different variables.

Symbol	Description
Q^N	Spreading sequence (traditional spread spectrum).
Q_s^D or Q^D	Spreading sequence segment.
$q(t)$	Spreading sequence (signal).
R_d, R_q	Signaling rate of the data and spreading signals, respectively.
R_m	Data embedding rate (bit/block).
$r(t)$	Received jammed (noisy) signal.
S^N	Host data vector (array).
$\hat{S}^N$, $\widetilde{S}^N$	Approximations of host data vector and host local perceptual parameters, respectively.
T	Message duration (PCM systems).
T_d, T_q	Signaling durations of the data and spreading signals, respectively.
U^N	Adapted watermarking vector (array).
$\hat{U}^N$	The estimated adapted watermarking vector (array).
V^N	Coded message vector.
W^N	Total watermarking vector (array).
$\hat{W}^N$	Estimated watermarking vector (array).
W_s^D, or W^D	Watermarking vector segment.
$\hat{W}_s^D$	Estimated watermarking vector segment.
$\widetilde{W}_s^D$ or $\widetilde{W}^D$	The enhanced estimated watermarking vector segment.
X^N	Composite data vector (array).
Y^N	Corrupted composite data vector (array).
Z^N	Total induced error vector measured at the receiver.
α^N	Perceptual adaptation scaling vector.
$\hat{\alpha}^N$	Approximated perceptual adaptation scaling vector.
$\widetilde{\alpha}^N$	Pre-equalized perceptual adaptation scaling vector.
β	Global power scale.
$\Phi_N(y^N, k^N)$	Overall decoding function.

Abbreviations

AC	Alternative current (used to refer to nonzero-frequency components of signals).
BER	Bit error rate.
DC	Direct current (used to refer to zero-frequency or average components of signals).
DC-QIM	Distortion compensated-quantization index modulation.
DCT	Discrete cosine transformation.
DFT	Discrete Fourier transformation.
DS	Direct sequence.
DSSS	Direct sequence spread spectrum.
FH	Frequency hopping.
JPEG	Joint photographic experts group.
LSB	Least significant bit.
LSBM	Least significant bits modulation.
MPEG	Motion pictures experts group.
MSE	Mean squared error.
PCM	Pulse code modulation.
QIM	Quantization index modulation.
SNR	Signal-to-noise ratio.
WNR	Watermark-to-noise ratio.

Contents

CHAPTER ONE

General Introduction

1.1 Introduction:

Although many *information hiding* techniques had been used throughout the history for *concealing the existence* of secret information within innocuous messages, it has been only a few decades since this subject has become a research topic*. At that time, most of the efforts were focused on the copyright protection applications, and they were primarily motivated by the video, audio and related materials (well known today as multimedia) production companies [1].

However, information hiding didn't receive much attention from the academic research community until the beginnings of the 1990's, which have witnessed a major development of many new well accepted multimedia data standards those have made digital multimedia data more widely used and more easily integrated into computer applications. The ease with which digital data is shared, copied and manipulated has motivated the rapid development of many of the applications of information hiding since then, especially, digital copyright protection applications well known in the literature as *digital watermarking*[**], and the covert communications applications known as *steganography*[***].

Many new hiding techniques and models have been proposed by the authors and researchers during the last decade, in addition to those been adopted or extended from the other fields of science like communications theory, channel coding theory, information theory and even cryptography.

* This can be traced back into a patent that has been granted at 1954 to Emil Hembrooke of the Muzac Corporation, entitled "Identification of sound and like signals" in which there is a description to a method for imperceptibly embedding an identification code within the music for the purpose of proving ownership. The patent states that "it can be likened to a *watermark* in paper" [1].

** The keyword "watermarking" is strongly related to the paper watermarks those have appeared since the end of the thirteenth century and have been used to differentiate among paper makers [2].

*** The keyword "steganography" has Greek roots and it literally means "covered writing" [2, 3, 4, 5].

The provision of a number of the supporting tools (such as the human perception models) -on the other hand- has aided the improvements made by some authors to the existing hiding techniques. Many articles, letters, technical reports and studies about this subject have been published during the last decade and are forming its basis [2].

Today, digital copyright protection is planned to become an integral part of the forthcoming still image compression standard JPEG2000 and the video compression standards MPEG-4 and MPEG-7 [6, 7, 8]. This will urge research groups to start cooperating to build a more unified theory about the subject, and would eventually lead to adopt some watermarking standards.

On the other hand, steganography is expected to become an additional security tool that can be used in combination with cryptography -as it is not expected to replace encryption algorithms been in use for a long time- to achieve higher rates of security. However, this hasn't changed the focus of many authors to develop hiding techniques those are highly secure by their own*. In addition to these applications, many other areas today are exploiting information hiding technologies.

Although none of the hiding techniques those have been proposed so far has yet met all the expected requirements of its corresponding application, several commercial software packages (which have employed some novice hiding techniques) have been developed during the last few years. Examples for these software packages include the image domain steganographic packages: StegoDOS, White Noise Storm and S-Tools [3, 4], and the watermark embedding package Digimarc [9] by Digimarc Corporation (which has become an integral tool of the Adobe Photoshop image editor).

* It will be seen later that the steganographic security requirements are somewhat different from the cryptographic security requirements.

1.2 Terminology:

This section is intended to introduce the common keywords those will be used throughout this thesis. Here, the information hiding problem will be viewed as a *communications channel* that is to be achieved between the *encoder* (the *information hider*) and the *decoder* (the *receiver*) despite of the existence of an *opponent* (the *attacker*[*]), who represents both *intentional* and *unintentional threats* to the security of that channel.

In a general information hiding context, the information (*message*) that is to be hidden will be referred to as the *embedded data*, the data into which this message is to be embedded will be referred to as the *host data*, while the corresponding resultant of *embedding* the former message within the latter host data will be denoted as the *composite data*. However, in the digital copyright marking context, the more common terms in the watermarking literature will be used to avoid confusion. Here, the embedded data will be referred to as the *mark* or the *watermark*, the process of embedding this mark will be referred to as the *marking process*, while the corresponding composite data will be denoted as the *marked data*. Usually, marking techniques involve -also- some form of a *key* that controls the marking process and restricts the access to the mark.

Different terms are used by authors in steganographic literature [2]. However, they will not be used throughout this text, and are mentioned here for completeness of this terminology. In steganography, the host data is usually referred to as the *cover data* or *cover-text*, while the *stego data* (or *stego-text*) refers to the composite data (by analogy to the *cipher-text* term used in cryptography). The key that controls the *hiding process* and restricts the access to the *secret message* is referred to as the *stego-key*.

* The term "attacker" here will be used to refer to any third party that aims at breaking the *secrecy* of the communications channel (*passive adversary* scenario) in the steganographic context, while it will be used to refer to third parties aiming at breaking its integrity (*active adversary* scenario) in the digital marking context [10, 11].

1.3 Information Hiding Applications:

Many information hiding applications exist today. These applications all share the desire to embed some extra information -with the least possible impact- within the host data object, which can be an image, an audio or a video signal*. However, these applications differ in the requirements they impose, and hence, in the techniques used to achieve them. Accordingly, information hiding applications can -generally- be classified into [1, 12]:

- ***Covert communications (steganographic) applications:*** those aim at concealing the existence of secret messages [3, 5, 9, 13] within innocent cover data in order to achieve covert communications between two parties. That is, in steganographic applications -as opposed to cryptographic applications- a successful attack consists in detecting the existence of this communication rather than detecting its contents [2]. This makes the *imperceptibility*** of the hidden secret message the main requirement in steganography.
- ***Copyright marking (watermarking) applications:*** those aim at embedding some owner identification information (which can be used later to prove ownership of the work in court) in the host data where it is required to permanently reside [6, 14] or at least survive any distortions applied to the marked data those do not render its content useless. Thus, a successful attack on a marked data consists in removing any trace of the embedded mark from it without severely degrading the perceptual quality of the host data [6]. So, the embedding of a *robust* mark is what watermarking all about.

* Other host objects include: text, free disk space, network packets … etc. [3].

** Imperceptibility of the embedded data is usually determined according to some perceptual measure depending on the type of host data under consideration (audio, still image, video … etc.). However, in steganographic applications, an additional requirement is to achieve imperceptibility against *steganalytic* attacks that aim at detecting the presence of hidden messages using such tool as statistical analysis.

- ***Transaction tracking (fingerprinting) applications:*** in these applications, a unique mark is embedded within each distributed copy of the host data. This mark contains such information as a serial number (a *fingerprint*) that identifies the legal recipient (the user) of that copy of data, which can be used to trace the source of any later illegal redistribution back to the corresponding user [1, 2, 6]. Fingerprints (as marks) have the requirement that they should be robust against modifications. However, the distribution of many differently marked copies of the host data imposes an additional threat, since now a number of users can combine their copies to destroy the embedded fingerprints (*Collusion* attack) [15, 16].
- ***Tampering detection (fragile marking) applications:*** a *fragile mark* is a one that is destroyed as soon as the marked data is sufficiently modified [2, 9]. This mark with a certain degree of robustness -as it has to survive such simple modifications as file format changes … etc.- can be used to authenticate the integrity of the underlying host data. So, a successful attack on a fragile mark consists in altering the contents of the host data containing it, without destroying that embedded mark [8].
- ***Seamless upgrading and data annotation applications:*** in this class of information hiding applications, additional enhancement or annotation data is embedded within the original data stream without imposing much degradation onto it, such that an upgraded receiver that is aware of this additional data can decode both the original and embedded data streams, while older receivers can still decode the original data stream without noticeable reduction in quality [17]. The main requirements of these applications are: the high embedding capacity and imperceptibility of the embedded data (not necessarily in the secrecy sense).

1.4 Literature Survey:

In this section a brief review of some of the techniques those have been proposed during the evolution of the field of information hiding will be introduced. The simplest and most direct hiding technique to start with is known as the least significant bits modulation (LSBM) technique. This technique simply replaces the L least significant bits (LSBs) of the host data samples with the bits of the data to be hidden. This way, both high embedding capacity and high imperceptibility level can be achieved. However, this technique tends to be very fragile to any applied manipulations. A variable-sized variant of this technique by Lee *et al*. [18] varies (adapts) the number of LSBs those are to be used for embedding for each pixel within the host image according to the maximum difference among the values of pixels around it. To reduce the error caused by embedding, the least unaltered bit of each modified pixel is properly adjusted. The remaining error is then diffused throughout the neighboring pixels of the image to remove contouring artifacts.

Zhao *et al.* [19] have proposed to embed watermark bits within triples of coefficients pseudorandomly selected from the mid frequency bands (as coefficients in this band have moderate variances usually, such that smaller changes would be sufficient) of the JPEG-like 2-dimensonal discrete cosine transformation (DCT) of the nonoverlapping 8*8 image blocks. With the predefined table of valid "1" and "0" bit representations being available, a watermark bit is embedded within a quantized coefficients triple by changing the relative magnitudes of the coefficients to reveal the closest corresponding relationship in the table if the required change would not exceed some predefined value, else, the triple would be labeled as "invalid" by changing it to the closest invalid triple representation in the table to limit the embedding induced distortion.

"Patchwork" an image watermarking technique by Bender *et al.* [20] embeds the watermarks within the statistics of host images. That is, to embed a single bit here, N pairs of nonoverlapping regions on the host image are pseudorandomly chosen. Then, the luminance level of one region of each pair is increased by a small amount δ, while the luminance of the other is decreased by the same amount so to increase the difference between the two luminance levels in each pair by 2δ. Now, with the statistical fact that the sum of differences of any randomly chosen N pairs of regions should be around zero for large N, it is possible to assert whether an image has been marked using a given key or not depending on whether the corresponding sum of differences is around 2δN or not.

Cox *et al.* [14] have proposed to encode watermarks as pseudorandom noise-like sequences those are to be *spread* over the perceptually most significant *spectral* components of the host images, where common signal processing operations are expected to impose the minimal modifications. Here, the watermarking sequence is added to the N lowest frequency coefficients (excluding DC) of the 2-dimensional DCT of the image after linearly scaling (adapting) the watermark elements to these DCT coefficients, (each element to its corresponding DCT coefficient). The inserted watermark sequence is retrieved from the corrupted marked image by simply subtracting the original host image (required to be available to the decoder) from the former one and then reversing the adaptation process (with the aid of the available host data, which would enable perfect recovery from the effect of the imposed *adaptation distortion* on the embedded mark). The decoder then correlates the resulting sequence with a regenerated version of the watermark to decide whether they match.

A spatial domain variant of the *spread spectrum* hiding technique has been proposed by Marvel *et al.* [21]. Here the message bits are modulated

with some pseudorandomly generated noise sequences and are then inserted into the image within the spatial domain rather than the transformation domain. To extract the embedded message, the receiver first tries to find an approximation to the original host image by filtering the incoming image using the *space-variant* adaptive Wiener filter. This filter takes advantage of the knowledge about the statistics of the embedded data (the pseudorandom noise statistics) and the local statistics of the input images to accordingly adapt its parameters in order to determine a minimum mean square error approximation of the original host image from the incoming -noisy- one [22]. This approximation is then to be subtracted from the incoming image to find an estimate of the embedded sequences those would be decoded for the message bits using a correlation-based matched filter, later.

Fridrich [12] -on the other side- has proposed to encode watermarks as *smooth* 2-dimensional spatial patterns with larger pseudorandomly shaped homogeneous patches (compared to highly freckled noise) to ensure that they would reside in the low frequency (perceptually significant) components of the image so that higher robustness would be achieved. An embedded watermark is extracted by subtracting the original image (available to the decoder) from the marked image first, and then correlating the resulting estimate of the embedded pattern with a regenerated version of the watermark to decide whether they match.

Fridrich [12] has also proposed to use some pseudorandom orthogonal transformation (a projection onto a set of orthogonal basis vectors calculated here by applying the Gramm-Schmidt procedure to orthogonalize the pseudorandom vectors generated using some secret key) instead of the standard DFT (discrete Fourier transformation), DCT or wavelet transformations to increase the system security. The watermark is inserted into the transformation coefficients of the host

signal in a similar fashion to that of [14]. To check for the presence of some watermark, the suspected signal is first transformed using the key-dependent basis vectors and then the resulting transformation coefficients are correlated against the given reference watermark.

Chen and Wornell [23, 24, 25] have proposed to embed data using ensembles (sets) of nonlinear functions (called scalar *quantization* functions) with disjoint output ranges (quantization cells) and near unity transfer functions (e.g. staircase like transfer functions). Here, the message (to be embedded) is to *modulate the index* of (select) the quantizer to be applied to the host signal. So, to decode a message that has been embedded with the *quantization index modulation* (QIM) scheme above, the decoder needs to simply determine the index of the quantizer (or the quantization cell) corresponding to the received signal value. Distortion compensated-QIM (DC-QIM) is a modification to the original scalar quantization embedding scheme with the former quantization error being scaled by (1-α), as a portion α ($0 < \alpha < 1$) of the original host signal is added back after quantization to reduce the embedding error, such that the quantization cells can be scaled by ($1 / \alpha$) to increase the system robustness without increasing the induced error.

An image-adaptive uniform scalar quantization based technique that employs erasure and error correction codes to deal with the synchronization problems (resulting from host-adaptation) has been proposed by Solanki *et al.* [17]. Here images are segmented into nonoverlapping 8*8 blocks, with the data being embedded into the low frequency DCT coefficients of these blocks. In the first version of the proposed technique there, a large nonbinary alphabet (an ensemble of much more than two scalar quantizers) is used. An alphabetical symbol is embedded (through applying the corresponding quantizer) within the low frequency coefficients of an entire block if its energy exceeds some given

threshold, otherwise the block is not used for embedding and is considered as if it has been *erased* at the encoder. The second proposed technique, however, selectively embeds the data bits within the individual DCT coefficients rather than DCT blocks. Coefficients those have higher energy than the threshold are used for embedding, while the others are erased at the encoder. Using an error correction code with an appropriate rate for the techniques above, the decoder would be able to survive errors due to given attack distortions.

Fridrich [12] -on the other side- has proposed to use an ensemble of nonuniform scalar quantizers to increase system security. The quantization cells here are defined according to a geometrical sequence with some secret quotient that should -generally- ensure a monotonically increasing quantization cell sizes. This way, the quantizer index (or equivalently the embedded message) corresponding to some received composite signal sample is kept secret to those adversaries whom are unable to regenerate the exact geometric sequence used for embedding.

An interesting proposal to combine both spread spectrum and uniform scalar quantization based embedding techniques has been made by Fei *et al.* [26]. The embedding here is to take place in the DCT domain representation of the 8*8 image blocks, with the equally indexed DCT coefficients (coefficients with the same frequency) through the image blocks being considered a subchannel of the overall embedding channel (this means that there exist 64 such subchannels, corresponding to the different 64 DCT coefficients in each block). Fei *et al.* have proposed to build some selection table with an entry that reveals whether a spread spectrum or a quantization based technique is to be used to embed data within each subchannel. The entries of this table are determined according to some objective measure of the robustness of each technique

within each specific subchannel (such as the expected correlation coefficient for the extracted watermark against the original one).

1.5 The Aim of this Thesis:

In this work, the traditional spread spectrum watermarking technique is to be modified, so that the common scaling-based host-adaptation schemes can be reformulated in such a way to assure that the modified adaptation processing would not induce any distortion -by itself- into the embedded data, hence, completely relieving the embedded watermarks from this additional source of distortion, and most importantly enabling the modified watermarking system to freely choose among the perceptual adaptation models, without worrying about the ability of equalizing the correspondingly induced distortions (into the embedded watermarks), the matter that could be of great influence on the whole watermarking problem. The proposed modifications are then to be implemented and tested (using MATLAB 6.5 script language) to practically approve the advantages gained here and to make sure that these proposed modifications have no disadvantageous impacts on the robustness of the resulting system.

1.6 Thesis Organization:

After this brief review of the general terminology and concepts of the subject, the basic theory underlying this field of technology is to be introduced. The general model of the information hiding problem as a communications channel with side information will be described in section 2.2. The characteristics of this model are to be discussed in terms of the rate-distortion-robustness tradeoff in section 2.3 and the most important classifications of the information hiding systems in section 2.4. The watermarking problem will be further explored along with the related

security/threats issues in section 2.5, while sections 2.6 and 2.7 will be devoted to review the general spread spectrum coding technique and watermarking model, respectively. The second chapter will be concluded in section 2.8 with a brief discussion about some practical issues concerning the robustness-imperceptibility tradeoff.

The proposed modified spread spectrum watermarking system is to be first introduced in chapter three. In section 3.2, a short review of the new concepts and modifications to be applied to the traditional spread spectrum watermarking model of sec. 2.7 will be provided, while the proposed image watermarking model is to be developed in section 3.3 after. Chapter three will be concluded with a comprehensive experimental examination of the performance of the proposed system in section 3.4.

In chapter four, some important enhancements to the host-adaptation section of the proposed system -presented in chapter three- are to be made. A detailed description to these enhancements will be provided first in section 4.2, after which two suggested implementations of these enhancements are to be experimentally examined in section 4.3.

Concluding remarks and suggestions for future work will be given in chapter five.

CHAPTER TWO

Theoretical Background

2.1 Introduction:

Perhaps it has been the development of the first models of the information hiding problem during the beginnings of the 1990's that has led to the subsequent rigorous understanding of this field of technology. This has been the first step of evolution of the earlier literature about the subject that had mostly focused on describing novice information hiding algorithms and techniques before.

The first model came in the early 1990's, when the information hiding problem was modeled -for the first time- as a communications channel, with the host data and any subsequent attack distortions introduced to the composite data been treated as noise. The overall perceptual fidelity was retained by globally constraining the embedding power.

However, in 1999, it has been recognized that the information hiding problem would have been more accurately modeled by a communications channel with *side information*, as is described next [1, 6].

The reader is referred to Appendix A for notation conventions used.

2.2 Statement of the Problem:

In what follows a generic model of the information hiding problem is described [6]. Suppose that there exist a host data source producing random variables $\mathbf{S} \in \mathcal{S}^N$ with a probability distribution $p_{\mathbf{S}}(\boldsymbol{s})$, a side information source producing random variables $\mathbf{K} \in \mathcal{K}^N$ with a probability distribution $p_{\mathbf{K}}(\boldsymbol{k})$, and a message source producing random variables $M \in \mathcal{M}$ with a uniform distribution over the message set $\mathcal{M}$:

- $\mathbf{S} = S^N = (S_1, S_2, \ldots, S_N)$ is the block (vector) of host data where the message M is to be embedded. It could be a block of data samples in the time or spatial domain, a block of transformation domain coefficients or a projection of any of them.

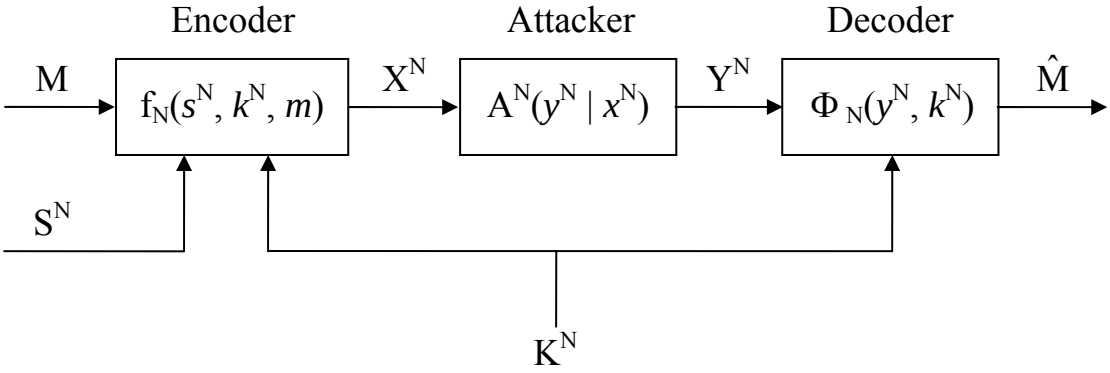


Fig. 2.1 A generic model of the information hiding problem.

- **K** = $K^N = (K_1, K_2, \ldots, K_N)$ is the side information that is available to the encoder and to the decoder but not to the attacker. The side information **K** plays two roles here. First, it may be some form of a cryptographic key (e.g. to a pseudorandom codes generator) that is necessary for encoding and decoding the message. This type of side information is independent of the host data. Second, it may provide some information about the host data, such as its structure, properties, hash values or it may provide complete information about the host data, in which case, **S** should be a deterministic function of **K**. Generally, the dependencies between **S** and **K** can be modeled by the joint probability distribution $p_{\mathbf{S},\mathbf{K}}(\boldsymbol{s}, \boldsymbol{k})$.
- $M \in \mathcal{M}$, is the message that is to be embedded in the host data at a rate of R_m bit per block, where $R_m = \log_2(|\mathcal{M}|)$.

The information hider (see Fig. 2.1) applies the embedding function f_N to the host data S^N, the side information K^N and the message M to produce the composite data X^N that should be perceptually similar to S^N.

Next, the attacker is expected to pass the composite data X^N into a random attack channel $A^N(y^N \mid x^N)$ that produces the *corrupted composite data* Y^N in an attempt to force the decoder to make an unreliable estimate of the embedded message. The channel $A^N(y^N \mid x^N)$ may represent either a

deterministic attack channel of the form $y^N = g_N(x^N)$, where the corrupted composite data Y^N is a function of X^N (here, $A^N(y^N \mid x^N)$ is a deterministic map from $\mathcal{X}^N$ to $\mathcal{Y}^N$ with the only possible values being ones and zeros), or it may represent a *probabilistic attack channel* that could (sec. 2.5.1) be represented by additive random noise insertion: $y^N = x^N + j^N$, where J^N is an additive noise signal with a probability distribution $p_{\mathbf{J}}(j^N)$, such that the dependencies between X^N and J^N can be modeled by $p_{\mathbf{X},\mathbf{J}}(x^N, j^N)$.

To decode the embedded message, the receiver applies the decoding function Φ_N to the corrupted composite data Y^N and the available side information K^N to produce an *estimate* of the embedded message $\hat{M}$.

The capacity of the hiding channel above can be described [23] as the total amount of information that can be transmitted through the channel given the side information available while decoding $I(\mathbf{X};\mathbf{Y} \mid \mathbf{K})$ minus the amount of information allocated to the host data given the side information available while encoding $I(\mathbf{X};\mathbf{S} \mid \mathbf{K})$, or symbolically [6]:

$$C = \max_{p(\boldsymbol{x}|\boldsymbol{s},\boldsymbol{k})} \min_{A^N(\boldsymbol{y}|\boldsymbol{x})} [I(\mathbf{X};\mathbf{Y} \mid \mathbf{K}) - I(\mathbf{X};\mathbf{S} \mid \mathbf{K})] \tag{2.1}$$

where, the inner minimization of the channel capacity over all the expected attack channels $A^N(y^N \mid x^N)$ those obey the maximum allowable attack induced distortion constraint (D_2) corresponds to the compromise with the channel capacity that is necessary to achieve reliable communications despite of these attacks, while the outer maximization over all the distributions $p_{\mathbf{X}|\mathbf{S},\mathbf{K}}(x^N \mid s^N, k^N)$ corresponds to employing the best embedding statistics obeying the maximum allowable embedding induced distortion constraint (D_1) to maximize the channel capacity, as:

$$\sum_{m \in \mathcal{M}} \sum_{k^N \in \mathcal{K}^N} \sum_{s^N \in \mathcal{S}^N} \frac{1}{|\mathcal{M}|} \cdot p_{\mathbf{S},\mathbf{K}}(s^N, k^N) \cdot d_N(s^N, x^N) \le D_1 \tag{2.2}$$

$$\sum_{x^N \in \mathcal{X}^N} \sum_{y^N \in \mathcal{Y}^N} A^N(y^N \mid x^N) \cdot p_{\mathbf{X}}(x^N) \cdot d_N(x^N, y^N) \le D_2 \tag{2.3}$$

with $d_N(x^N, y^N)$ being some distortion measure that estimates the amount of perceptual irrelevance of y^N to x^N, and that the probability distribution $p_{\mathbf{X}}(x^N) = \sum_{\mathcal{M}} \sum_{\mathcal{K}^N} \sum_{\mathcal{S}^N} p_{\mathbf{X}|\mathbf{S},\mathbf{K},M}(x^N \mid s^N, k^N, m) \cdot p_{\mathbf{S},\mathbf{K}}(s^N, k^N) / |\mathcal{M}|$. Note here that the attack induced distortion -generally- depends on the composite data through $p_{\mathbf{X}}(x^N)$ and $d_N(x^N, y^N)$. However, for the case of independent additive noise attack channels (when $p_{\mathbf{X},\mathbf{J}}(x^N, j^N) = p_{\mathbf{X}}(x^N) \cdot p_{\mathbf{J}}(j^N)$, so that: $A^N(y^N \mid x^N) = p_{\mathbf{X}+\mathbf{J} \mid \mathbf{X}}(x^N + j^N \mid x^N) = p_{\mathbf{J} \mid \mathbf{X}}(j^N \mid x^N) = p_{\mathbf{X},\mathbf{J}}(x^N, j^N) / p_{\mathbf{X}}(x^N) = p_{\mathbf{J}}(j^N)$) with distortion measures d_N those are functions of the induced *error* only (e.g. a mean squared error function), the distortion induced by the channel: $\sum_{\mathcal{X}^N} \sum_{\mathcal{J}^N} p_{\mathbf{J}}(j^N) p_{\mathbf{X}}(x^N) \cdot d_N(j^N) = \sum_{\mathcal{J}^N} p_{\mathbf{J}}(j^N) \cdot d_N(j^N)$, would depend on the additive noise only. Also note that $\mathcal{S}^N = \mathcal{X}^N = \mathcal{Y}^N$, as the distortions induced through any part of the channel (described by equations 2.2 and 2.3) may not change the format of the host data.

2.3 Rate-Distortion-Robustness Tradeoff:

It has been mentioned earlier that there are generally three main parameters (requirements) that characterize the different applications of information hiding, namely: the embedding capacity (rate), the host perceptual fidelity (inversely related to the embedding induced distortion) and the immunity (robustness) of the embedded data to intentional and/or unintentional manipulations. Generally, an information hiding system is required to provide a high embedding capacity and a high robustness to all expected active attacks with a low impact on the perceptual fidelity of the host data. However, due to the contradictory nature of these requirements (as revealed by equations 2.1 – 2.3), it tends to be an impossibility to *simultaneously* achieve them all. Fig. 2.2 shows the contradiction relations between capacity (rate), fidelity (inversely related to distortion) and robustness parameters from the following view point.

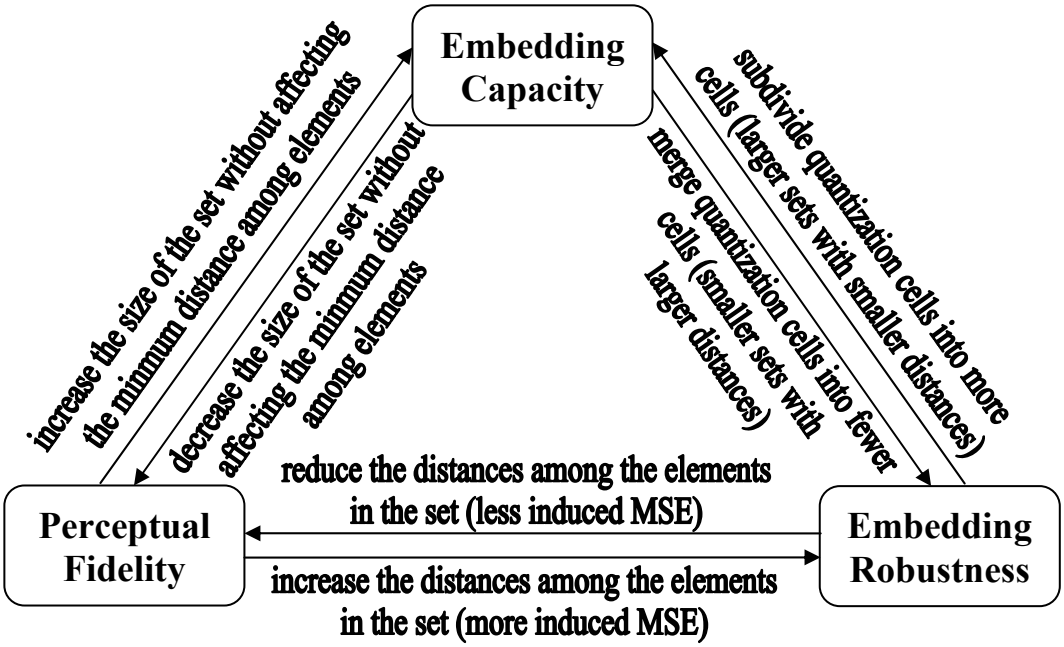


Fig. 2.2 Rate-Distortion-Robustness tradeoff. The arrows above represent the contradiction relationships between the pairs of parameters they connect.

That is, if the message to be embedded is represented by an element that is randomly selected from a finite set of quantized values (or vectors, the important aspect here is that there is a finite distance between any pair of these distinct elements), then from this view, it can be seen that the amount of embedded information (the capacity) depends on the size of the set (the number of distinct elements in the set), the amount of embedding induced distortion depends on the mean squared error (MSE) induced by embedding this message (or the mean of squares of the distinct elements in the set assuming a uniform distribution over the message set), while the robustness of the embedded data depends on the minimum distance between any given pair of elements in the set.

In practice, however, it is not necessary to simultaneously achieve all these requirements, since that each distinct information hiding application has its prioritized view of performance optimization. So practically, a better information hiding technique is a one that provides a more controllable rate-distortion-robustness tradeoff in terms of the prioritized parameters from the viewpoint of its corresponding application domain.

2.4 Classes of Information Embedding Methods:

Information embedding methods can -generally- be categorized according to several aspects, such as the side information available to the encoder and decoder, the host signal interference rejection properties, the type of constraints imposed to ensure perceptual fidelity, the host-adaptability properties, and so on. Embedding methods will be first classified according to their host-adaptability properties in here:

- ***Host-adaptive embedding methods:*** they have the property that they *utilize* their knowledge about the *local properties* of the host data to locally vary the embedded data accordingly [1]. Varying the embedded data includes locally adjusting the amount of embedding power and/or controlling the locations where embedding is to take place (or generally adjusting the *embedding maps*). These methods are usually superior to nonadaptive embedding methods in terms of the overall perceptual fidelity, embedding robustness and/or capacity due to their ability to utilize the *perceptually* significant and/or redundant components within host signal more efficiently. However, these methods require some information about host data to be available while decoding so that the embedding maps can be perfectly regenerated. On the other hand, host-adaptive embedding methods where decoders extract these required -usually sensitive- information from the received -possibly corrupted- composite data suffer from many synchronization problems, due to the inability of decoders to make perfect estimations of the original embedding maps using modified versions of the host data (the implications of this problem on host-adaptation will be discussed in chapter 4).

  Host-adaptive encoders can generally be described by the model shown in Fig. 2.3. First, the message is coded into V^N by applying

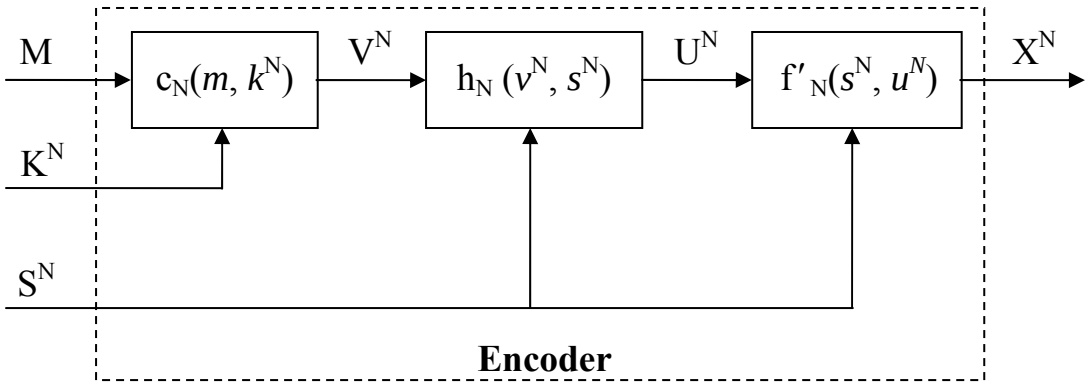


Fig. 2.3 A generic model of host-adaptive encoders.

the coding function c_N to both the message M and the side information K^N, where c_N may be an error control coding function and/or a modulating function with some random modulation code. The coded message V^N is then adapted to the host data* S^N as they pass through the *host-adaptation function* h_N. It is the host-adapted coded message U^N -in combination with the host data S^N- that is passed then to the embedding function f'_N (which plays the role of a code multiplexing function) to produce the composite data X^N.

- ***Host-nonadaptive embedding methods:*** these embedding methods *do not utilize* their knowledge about the host local properties to vary the embedded data accordingly. However, these methods can still utilize the *global* properties of the host data to control the global embedding parameters. Host-nonadaptive embedding methods are generally described by the model shown in Fig. 2.4. As is shown, no host-adaptation is applied to the coded message V^N before being passed to the embedding function f'_N (note that K^N includes cryptographic keys and global information contents).

Decoding methods -on the other hand- can be categorized according to the side information available to the decoder into:

* It should be emphasized though, that the information available to the encoder about the host data (here) takes the form of side information according to the adopted model of the hiding problem. However, to clarify the idea here, the host data (S^N) itself (rather than the side information K^N) has been passed into the host-adaptation function h_N.

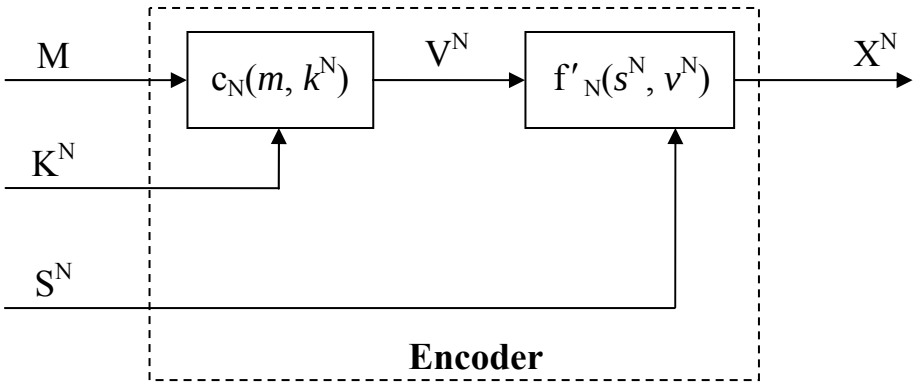


Fig. 2.4 A generic model of host-nonadaptive encoders.

- ***Informed decoding:*** informed decoding methods require some side information about the host data to be available while decoding to achieve the necessary synchronization with the encoding end, in order to recover the embedded data*. In one common informed decoding scenario, the complete host data is available to the decoder (in watermarking literature, these scenarios are usually called *private marking*). Other scenarios include cases where only partial information about the host data is available to the decoder.

  Although these methods are usually superior to non-informed decoding methods, the limitation that some side information needs to be -hypothetically- communicated between the encoding and decoding ends tends to limit their applicability domains.

- ***Blind decoding:*** blind decoding methods do not require any side information other than cryptographic keys (those are independent of the host data) to be available while decoding to recover the embedded data [6]. This includes the situations where no additional synchronization informations are required while decoding and the other situations where the decoders are responsible to estimate such informations from the received data. In the watermarking literature,

* There are situations where the availability of this side information to the decoder is necessary to achieve higher robustness, capacity or imperceptibility rather than being necessary for achieving synchronization, however, the decoders in these scenarios are still regarded as informed decoders.

these methods are usually called *blind watermarking* methods, in the sense that decoders are unable to see the original host data.

Applications here, span a wider range than that of informed decoding methods, as it is always possible -only from *applications* point of view- to remove the limitation on the availability of side information about the host data to the decoder.

Information hiding techniques used so far can generally be classified according to their host interference rejection properties into [23, 24, 25]:

- ***Host interference rejecting techniques:*** these techniques exploit the knowledge about the host data available while encoding to embed the data into in a way that makes its recovery independent of the former host* [23, 24, 25]. These techniques do not require any side information about the host data to be available while decoding (actually the availability of any information about the host data to the decoder provides no additional information about the embedded message), since that the embedded data resides in an independent code domain from that of the host data.

  Host interference rejecting techniques include such methods as: the least significant bits modulation (LSBM) methods, and quantization based methods such as the quantization index modulation (QIM) method introduced by Chen and Wornell [23, 24, 25].

- ***Host interference non-rejecting techniques:*** these techniques treat host signals as sources of unknown noise or interference without exploiting the knowledge about the host data available while encoding [23, 24, 25]. Here, data is embedded by linearly summing

* It should be emphasized here that exploiting the knowledge about host data in this context is totally different from the context of host-adaptability. Host interference rejection properties of information hiding algorithms are solely related to the code multiplexing problem, rather than the host-adaptation problem.

it with the host, which -in turn- would act as a source of additive noise with high (but limited) total power* that would interfere with the decoding process. To reduce the effect of this interference to a far degree, *spread spectrum modulation* (coding) techniques -well known in the field of *jamming resistant* communications- are employed. Here, spread spectrum techniques sacrifice the embedding capacity to increase the resistance of the embedded data to host and jamming interferences, the property that has made them more appropriate for robust marking applications. Although these techniques may not completely reject host interference under blind decoding strategies, they are still very robust against severe jamming attacks if so designed**. On the other hand, host interference can be completely removed (subtracted), if sufficient side information about the host data is available to the decoder (informed decoding). In these scenarios, spread spectrum techniques can prove to be perfectly robust.

Spread spectrum techniques -popularized to the community of information hiding by Cox, Kilian, Leighton and Shamoon [14]- have found great acceptance till now in the field of robust marking. Many variations of the spread spectrum model of [14] exist today.

Statistical embedding techniques -on the other hand- such as the technique called "Patchwork" -introduced by Bender, Gruhl, Morimoto and Lu [20]- belong to this class, too, since that they model host signals as sources of unknown noise with some expected statistical characteristics.

* Note here that host signals do not have infinite power spectrum as in thermal noise case, the property that is central to the work of jamming resistant techniques.

** Surviving the high energy host interference should necessarily imply that these techniques can also survive such severe attacks, since after all jamming energies cannot be more than a small portion of the host signal energy -if perceptual fidelity of the latter is to be conserved-.

2.5 Digital Watermarking:

As it has been defined earlier, a digital mark (watermark) is a special form of a message -usually as short as is required to include such information as owner identification code, time stamp … etc.- that is permanently embedded in the host data and is required to survive any manipulations applied to the marked data that do not render its contents commercially useless. Thus, unlike the case in cryptographic and steganographic applications, a watermarking technique is considered to be secure if and only if it can embed a watermark that is as robust as is defined above. However, watermarking systems still need to obey Kerckhoff's security principles below [11] to ensure that an adversary cannot take advantage of some of the weakness points in the system, so:

1. an embedded watermark should be -at least practically- unbreakable (the estimation of the embedded watermark through the analysis of the corresponding marked data should be impossible),
2. the security of the watermarking system should not depend on the details of the marking algorithm, rather,
3. the security of a given watermark should depend only on some key that is a *property* of its corresponding owner.

The definition above draws a very strict line between watermarking techniques those are *fully* robust and those are robust to some extent. However, since none of the watermarking techniques proposed so far has proved to be such robust, it has become a tradition to test the robustness of these techniques against some of the well-known attacks that a watermark may undergo. In the next subsection some of the well-known classes of attacks (which will be of a great help in understanding the heavy burdens imposed on watermarking techniques) will be addressed.

There has been an emphasis in watermarking literature (see ref. [22]) on the separation between the watermark *detection* process (represented by the decision made by the receiver on whether a specified mark has been embedded in a given data object or not based on the result of *correlating* the suspect watermark extracted from that object with the reference mark) and the watermark *decoding* process (that aims at recovering the information -or the message- coded within the watermark embedded in the data object). However, this text will focus on the more general process of watermark decoding. It is argued here that every watermark is a coded message and that the detection process is no more than deciding whether the message decoded from the recovered watermark matches the reference message under consideration or not. Watermarks those do not modulate any specific messages (as in the watermark structure of [14], which consists of a sequence of pseudorandom noise) can be thought of as carrying a single information bit (each) that asserts whether the containing object has been marked using the corresponding *owner-identifying key* or not.

Decision (detection) mechanisms are usually built to achieve some predefined statistical characteristics in terms of the *detection probability* and *false positives probability* (defined as the probabilities of detecting the presence of the reference watermark within the data object in the case that it *has* and the case that it *has not* been marked with the aforementioned watermark, respectively) on the basis of a given correlator statistics. Since the design of these mechanisms belongs to the field of *hypothesis testing*, it will not be explored any further here.

2.5.1 Classes of Attacks on Watermarking Systems:

Unlike the situation in steganography, in watermarking systems the detection of the existence of an embedded watermark in a marked data is

not regarded as an effective attack by itself. Thus, an effective attack on a watermark should include some *active manipulation* with the marked data that seeks to render the embedded watermark useless [10] (or more generally, decrease the probability of detecting it [22]) without severely distorting the containing host. These attacks can be categorized into:

- ***Probabilistic Attacks:*** it is always possible -in practice- (although this may -theoretically- involve some loss with generalization) to model the error induced by these attack channels with additive random noise. These channels are characterized by the variances and shapes (uniform, Gaussian, Laplacian … etc.) of the probability distributions of the additive noise they insert. Attacks in this category do not seek to destroy or diminish the embedded watermarks themselves, but, to insert some interference signals those would reduce the reliability of the results of (hopefully randomize the results of) the decoding process. Additive noise channels may be further classified into:

1. *Dependent Additive Noise Channels:* the inserted noise through such a channel is adapted to the local properties of the channel input. The aim of this adaptation is to either try to maximize the power of the inserted noise under the perceptual fidelity constraint (hence, causing as much disturbance as is possible), or to synchronize (in the disturbance sense) the inserted noise to the embedded data -if the adaptation procedure of the corresponding embedding algorithm is known to the attacker- to maximize the effect of the inserted noise on the embedded watermark.

   The *nonstationary* behavior of these channels is generally modeled by the conditional distribution $A^N(y^N \mid x^N) = p_{\mathbf{J} \mid \mathbf{X}}(j^N \mid x^N)$ (since that $y^N = x^N + j^N$, which means that Y^N is a deterministic function of J^N in the existence of X^N), where X^N is the channel input, Y^N is the channel output and J^N is the dependent additive

noise signal that can be described as a function of both the channel input and some other parameter $Z^N \in \mathcal{Z}^N$, which controls its random behavior, or: $j^N = g_N(x^N, z^N)$, such that $p_{\mathbf{X},\mathbf{J}}(x^N, j^N) \neq p_{\mathbf{X}}(x^N) \cdot p_{\mathbf{J}}(j^N)$.

2. *Independent Additive Noise Channels:* the inserted noise through such a channel is independent of the properties of the channel input (except for its global power properties -probably-). The *stationary* behavior of this channel can -generally- be modeled by the probability distribution of the inserted noise $A^N(y^N \mid x^N) = p_{\mathbf{J}}(j^N)$, where J^N is the noise signal generated independently of the channel input X^N and Y^N is the channel output (see sec. 2.2).

- ***Deterministic Attacks:*** this category of attacks includes a wide range of procedural operations those usually aim at either removing or diminishing the watermark itself or distorting its containing medium in some form to confuse the decoding process [2]. These attacks can be further classified into [14]:

1. *Analytical Attacks:* such as the autocorrelation and steganalysis (statistical analysis) attacks those aim at estimating (and hence removing) the embedded watermarks by autocorrelating (or statistically analyzing) the marked data or parts of it looking for any possible redundancy within the inserted watermarks.
2. *Destructive Attacks:* those include such common signal processing operations as lossy compression (MPEG, JPEG … etc.), linear/nonlinear filtering, visual/auditory enhancements, histogram equalization, dithering and many other operations those aim at diminishing the underlying watermarks.
3. *Desynchronization Attacks:* they include geometrical distortion attacks (for video and imagery mediums) such as rotation, scaling, translation, torsion, cropping, re-sampling (analog/digital conversions) and many other spatial transformations, which aim at

ruining the synchronization between the encoder and decoder in order to confuse the watermark recovery process.

4. *Collusion Attacks:* when several users -having differently marked versions of the same host data- combine their own copies in order to get a different one that is free of the watermark of any of them. In video marking domain, however, a number of the adjacent frames those are taken from some slowly changing scenes of the same copy of the clip may be combined to remove the embedded mark [8].

Today, a number of freeware watermark testing and benchmarking packages are available. With the combinations of different attacks these packages apply, they can generally give an idea about the robustness of the different watermarking algorithms in the real world, where there are no limitations (other than the fidelity limitation) on the forms and sequences of attacks a watermark may undergo. StirMark (a well-known image watermark testing package) for example, applies a sequence of geometrical distortions, low frequency deviations and random noise insertion [2, 5, 9] that has proved to be able to defeat any designed watermarking algorithm until now.

2.6 Spread Spectrum Techniques:

Spread spectrum techniques [27, 28] are those modulation or coding techniques in which the signal to be communicated is spread over a bandwidth in much excess of the minimum -necessary to transmit it- by modulating it with a code (carrier) called the *spreading sequence* that is generated independently of the signal to be transmitted and synchronously with the corresponding receiver to despread it and recover the transmitted content. The idea behind spectrum spreading is that of *hiding* the power of a relatively narrowband signal within a much wider

band, such that only a small portion of the total signal power is added to each spectral component.

Spread spectrum techniques are usually aimed at achieving:

1. high resistance to jamming/interference signals with finite power,
2. low probability of interception/detection by an adversary, or
3. code division multiplexing, where the channels are made disjoint in code domain -rather than time or frequency domains- to minimize the cross interference among them and best utilize the resources.

Generally, spectrum spreading is achieved by either modulating the narrowband data signal with a *frequency hopping* (FH) carrier that causes pseudorandom frequency shifts to the narrowband signal, such that it would occupy a small pseudorandom fraction of the wider transmission band at any given time, or modulating it with a wideband signal -called *direct sequence* (DS)- that causes abrupt pseudorandom phase shifts to the narrowband signal in order to spread its spectrum, such that the entire transmission band is occupied at any given time. In what follows, the direct sequence scenario will be addressed as far as is necessary.

Referring to Fig. 2.5, suppose that there exists a data source producing a data signal d(t) (with a signaling rate of $R_d = 1/T_d$ message/second and an average power of P_d watt) that is to be communicated over the spread spectrum channel shown. To spread the power of this narrowband signal over the entire system bandwidth, the direct sequence spread spectrum (DSSS) transmitter multiplies the data signal by the wideband spreading sequence q(t) (with a signaling rate of $R_q = 1/T_q$ chip/second and a normalized energy over intervals of T_d seconds to preserve the power of the data signal at the transmitter output -for simplicity-) produced by seeding the random sequence generator with some secret key, such that:

$$\frac{B_q}{B_d} = \frac{R_q}{R_d} = \frac{T_d}{T_q} = G_p \qquad (2.4)$$

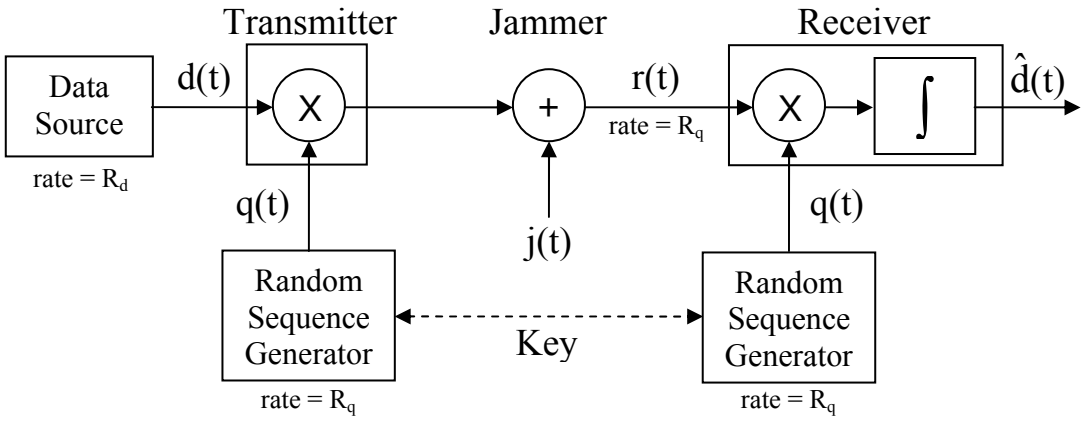


Fig. 2.5 A generic model for a direct sequence spread spectrum (DSSS) communications system.

where B_q and B_d are the DSSS system total transmission bandwidth and data signal bandwidth, respectively, and G_p is the *processing gain*.

Next, a possible adversary willing to achieve the maximum disruption to the communications (with a total jamming power of P_j watt) is expected to transmit his jamming signal j(t). The effect of this jamming signal takes the form of additive random noise with its power being either spread over the entire transmission band (broadband jamming scenario) or concentrated into some narrow band (narrowband jamming scenario).

At the receiving end of the channel, the incoming signal r(t) is a corrupted version of the transmitted signal with additive jamming noise (ignoring -for the time being- the effect of thermal noise):

$$r(t) = d(t) \cdot q(t) + j(t) \qquad (2.5)$$

To *despread* the spectrum of the received signal r(t), it is first multiplied by the spreading sequence q(t) regenerated at the receiver in synchronization with its input -in order to remove the effect of the abrupt phase shifts caused by the same sequence at the transmitter- before being passed into an integrator that accumulates the signal over boundary-synchronized intervals* of length T_d seconds to produce an estimate of the data signal d(t) at the receiver output $\hat{d}(t)$:

* Intervals those are synchronized to the time boundaries of received sequences.

$$\hat{d}(t) = \int_{T_d} r(t)\cdot q(t)dt = \int_{T_d} d(t)\cdot q(t)^2 dt + \int_{T_d} j(t)\cdot q(t)dt$$

$$= d(t) + \int_{T_d} j(t)\cdot q(t)dt \qquad (2.6)$$

where $\int_{T_d} q(t)^2 dt = 1$, since that q(t) is normalized over each interval of T_d seconds during which d(t) assumes a constant value. Note that the estimate produced at the receiver consists of the transmitted data signal (the first term) plus some *interference* signal (the second term) that actually is a processed version of the additive jamming signal inserted through the channel. To determine the variance (power) of this interference signal, the second term in equation (2.6) is examined. Here, the jamming signal j(t) is multiplied by the spreading sequence q(t), causing to spread its spectrum over the entire system bandwidth with a two sided power spectral density of $P_j/(2B_q)$, of which only a small portion would be allowed to pass through the integrator (see Fig. 2.6) that acts as a low pass filter with a bandwidth B_d corresponding to the interval T_d over which it accumulates the incoming signal (the total effect here is usually called *matched filtering*, corresponding to the direction of the matched sequence or vector over which the input signal is integrated). This means that the power of the interfering signal at the receiver output is reduced (by the aforementioned processing) into $P_j \cdot (B_d / B_q)$, making the signal-to-noise (signal-to-interference) ratio at the output of receiver:

$$SNR = \frac{P_d}{P_j}\cdot\frac{B_q}{B_d} = \frac{P_d}{P_j}\cdot G_p \qquad (2.7)$$

Equation (2.7) shows that spread spectrum techniques afford an opportunity to give a signal some desired power advantage over the other jamming/interference signals (which may be of much higher powers than the signal itself) with the appropriate band spreading. Again, notice that

these techniques are of no help in resisting thermal noise with infinite power spectrum (a constant strength over the infinite spectrum), where increasing the system bandwidth would *proportionally* increase the induced noise power through the channel and cancel the spreading effect.

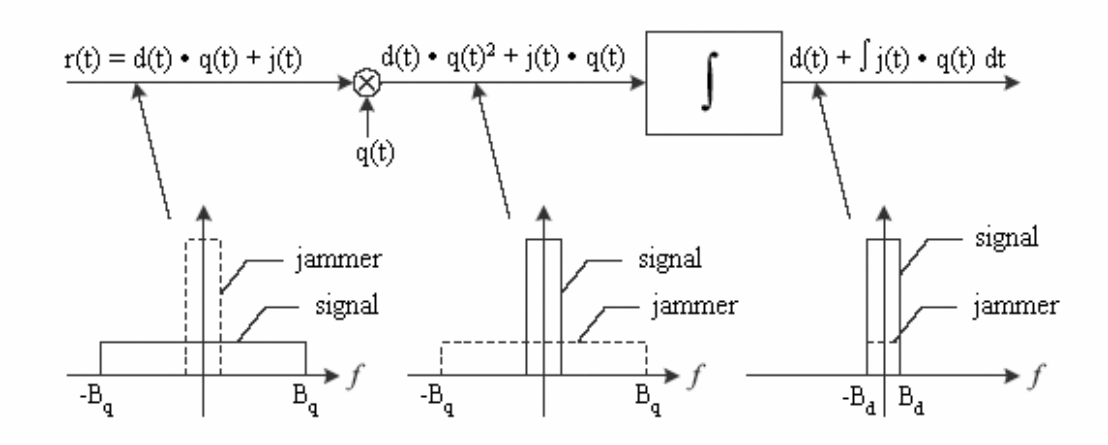


Fig. 2.6 The operation of a DSSS receiver [28]. Note here that the spectrum of the jamming signal shown on the left axes corresponds to a narrowband jamming strategy. However, all the other jamming strategies would -still- respond in the same manner -shown on the middle and right axes- to the DSSS receiver processing.

2.7 A General Spread Spectrum Watermarking Model:

In this section a general model for spread spectrum watermark embedding and decoding schemes is presented.

It will be assumed here that each watermark **W** encodes a message M selected out of a set $\mathcal{M}$ with $L = |\mathcal{M}|$ equally probable messages.

Fig. 2.7 [22] shows a generic model for a spread spectrum watermark embedding system. The spread spectrum encoder starts by seeding some pseudorandom generator with the key **K** (the cryptographic key content of the side information) to generate the spreading sequence **Q**. Here, the spreading sequence may be a single N-dimensional sequence Q^N that is to be modulated by the *quantized value* of the message M (the quantized value of the ith element -selected for embedding- from within the set $\mathcal{M}$ of quantized messages) through multiplying the latter by the sequence

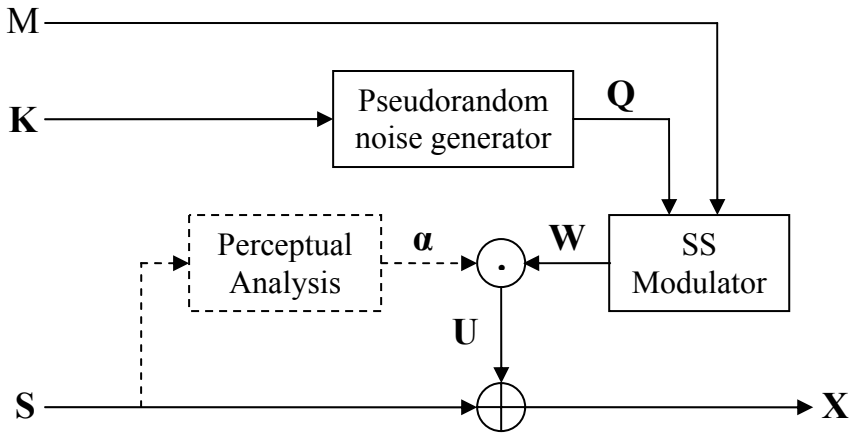


Fig. 2.7 A generic model of a spread spectrum watermark embedding mechanism. The dashed components here correspond to the optional embedding steps.

components to encode it within the resulting watermark W^N:

$$w^N = m_i \cdot q^N \mid M = m_i \tag{2.8}$$

In other scenarios, the pseudorandom generator may produce a set of L *orthogonal* N-dimensional sequences $\{Q_j^N \mid j = 1 \ldots L\}$ from which the modulator selects a single sequence Q_i^N for transmission according to the index i (with respect to $\mathcal{M}$) of the message to be embedded, hence, encoding the message within the *index* of the sequence, such that:

$$w^N = q_i^N \mid M = m_i \tag{2.9}$$

The resulting watermark W^N is then to be perceptually adapted to the local properties of the underlying host signal S^N to reduce the embedding impact on the perceptual fidelity of the latter. Here, the watermark W^N is scaled a component-by-component (as the dot in Fig. 2.7 shows) with some *scaling vector* α^N, whose components represent -according to the adopted perceptual model- the *imperceptibility scales** of the corresponding components of the watermark W^N in relation to the local properties of the host data. The adapted watermarking vector U^N is then added to the host signal S^N to form the marked data X^N:

$$x^N = s^N + u^N = s^N + \alpha^N \cdot w^N \tag{2.10}$$

* The scales those would make the effect of embedding the corresponding watermark components imperceptible -to a given extent- according to the adopted perceptual model.

To decode the message embedded within a marked -possibly corrupted- signal Y^N, the corresponding spread spectrum decoding system (shown in Fig 2.8) tries -first- to find an *estimate* of the embedded vector $\hat{U}^N$. Generally, there are two scenarios for finding such an estimate. The first scenario constitutes in using the available information about the host data -including the side information K^N and the received signal Y^N itself- to find some *approximation* to the host signal $\hat{S}^N$ that is to be subtracted from the received signal Y^N to reduce the host interference as much as is possible. Although this subtraction can significantly improve the calculated estimate of the embedded vector $\hat{U}^N$ (as in the technique proposed in [21]), it is still unable to completely remove host interference unless complete information about the host signal is available to the decoder (private marking) as is shown below (note the additional error term -enclosed by the parentheses- that due to the residual of the subtraction of an imperfectly approximated host signal $\hat{S}^N$):

$$\hat{u}^N = y^N - \hat{s}^N = (s^N - \hat{s}^N) + e^N + u^N \qquad (2.11)$$

where $e^N = y^N - x^N$, is the overall channel induced error (being considered from an additive point of view) with a probability distribution of:

$$p_{\mathbf{E}}(e^N) = \sum_{\substack{\{y^N \in \mathcal{Y}^N, x^N \in \mathcal{X}^N \\ | \, y^N - x^N = e^N\}}} p_{\mathbf{Y},\mathbf{X}}(y^N, x^N) = \sum_{\substack{\{y^N \in \mathcal{Y}^N, x^N \in \mathcal{X}^N \\ | \, y^N - x^N = e^N\}}} A^N(y^N \mid x^N) \cdot p_{\mathbf{X}}(x^N) \qquad (2.12)$$

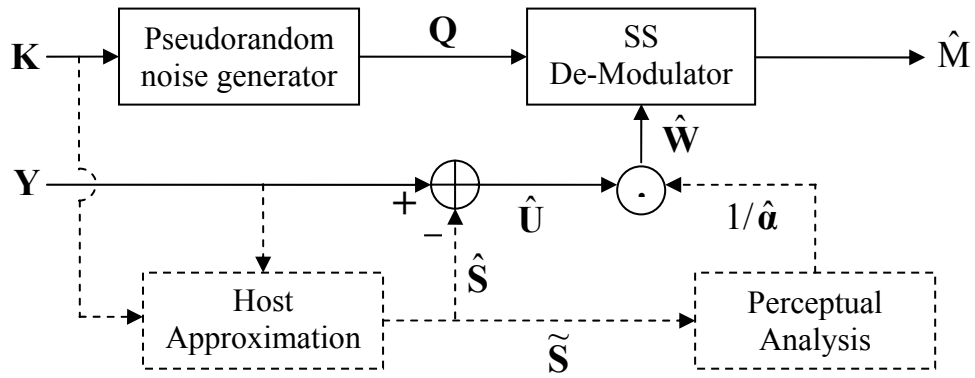


Fig. 2.8 A generic model of a spread spectrum watermark decoding mechanism. The dashed lines and boxes here correspond to the optional components of the decoder.

The second scenario, however, is to treat the received signal Y^N itself as an estimate of the embedded vector $\hat{U}^N$. Although this would include the entire host signal -instead of the smaller host approximation error- in equation 2.11, the received data Y^N is still the best possible estimate of $\hat{U}^N$ when host approximation is an unavailable option, so:

$$\hat{u}^N = y^N = s^N + e^N + u^N \tag{2.13}$$

An estimate of the embedded watermark $\hat{W}^N$ is then determined by trying to *equalize* (reverse) the effect of the perceptual adaptation process (if any has been applied at the encoder side) on the extracted estimate of the embedded vector $\hat{U}^N$. To do so, the decoder first seeks to find an appropriate approximation to the corresponding scaling vector (α^N) that has been used by the encoder, utilizing the available information about host signal. Note that this would require the determination of some host approximation* $\tilde{S}^N$ that is *necessary* for finding the approximate scaling vector $\hat{\alpha}^N$. The *componentwise* reciprocal of the approximated scaling vector $(1/\hat{\alpha}^N)$ is then to be multiplied a component-by-component by the vector $\hat{U}^N$ to determine an estimate of the embedded watermark $\hat{W}^N$.

Next, the decoder seeds the pseudorandom generator with the key **K** (required to decode the message that has been encoded using this key) to regenerate the corresponding spreading (or despreading) sequence **Q**. One out of two demodulation methods may be carried out here according to whether *value* or *index* modulation has been applied to encode the embedded message. In the first case, the regenerated spreading sequence Q^N is to be correlated with the estimated watermarking vector $\hat{W}^N$ to

* Note that the approximated host signal $\tilde{S}^N$ (needed to equalize the adaptation distortion) has been differentiated here from the previous approximation $\hat{S}^N$ to emphasize on the dependency of the adaptation process on the *local parameters* of the host signal rather than the host signal itself. This means that less information (or at least less details) about the host signal *may* need to be approximated here.

give an estimate of the embedded message $\hat{M}$, where:

$$\hat{M} = m_i \mid i = \arg\min_j \left\| < \hat{w}^N,\ q^N > - m_j \right\|, \text{(where j =1 … L)} \quad (2.14)$$

In the second case, however, all the regenerated sequences within the set $\{Q_j^N \mid j = 1 \ldots L\}$ are to be correlated with the estimated watermark $\hat{W}^N$ and the embedded message is assumed to correspond to the index i of the sequence Q_i^N that would produce the highest correlation coefficient, or:

$$\hat{M} = m_i \mid i = \arg\max_j < \hat{w}^N,\ q_j^N >, \text{(where j = 1 … L)} \quad (2.15)$$

Mathematically, the spread spectrum technique described here can generally be thought of as a method to achieve a secure communications channel by concentrating the whole signal power into one secret pseudorandom direction, that is, the direction of the spreading sequence Q^N (or Q_i^N). In this way, jamming signals (including host interference) with uniform power distributions over all N dimensions of the transmission space* would induce no more than 1/N (on the average) of their total powers into each distinct direction**. An important parameter here is the ratio between the dimensionality of the spreading sequences and the number of distinct messages (N / L), which corresponds to the processing gain of the embedding system. This parameter actually represents the achievable robustness gain in return of the corresponding sacrifice with embedding capacity when using a spread spectrum system with the given parameters.

* The space spanned by the N orthogonal basis vectors corresponding to the N orthogonal sequences of which the set of selected spreading sequences is a proper subset.

** Note that this means that L/N of the total jamming power would be induced into the L matched directions corresponding to the L distinct messages in the set $\mathcal{M}$ in the case of index modulation, while inducing only 1/N of the total jamming power into the only direction of the spreading sequence used in the case of value modulation. However, since that in this case the information is contained within the distances between the L different quantization levels assumed by these L different messages, the effect of the jamming power increases to L times its received level as these distances are -on the average- within 1/L of the original transmission power.

2.8 Watermark Embedding Domains:

In this section, the practical issues concerning the relationship between the watermark embedding domains and the robustness-imperceptibility properties of the embedding procedures are to be discussed. To begin with, it should be first stated that regardless of the specific domain where an embedding process is to take place, the robustness-imperceptibility properties of the embedded data are only dependent -*ideally*- on the perceptual significance-redundancy properties of the room -within the host signal- where it resides. However, due to the lack of existence of accurate models for human perception and the high computational complexity of the available models, in addition to the nature of traditional attacks those do not really comply to such models in basis, it has become evident that a better embedding scheme is a one that can *roughly* discriminate these significance-redundancy rooms -within the host data- on the basis of some host representation that is *well-matched* to the nature of these attacks, such that the *robustness* of the different components of the host signal to these attacks can be better defined.

This last observation has generally shifted the interests of authors in schemes where embedding takes place in time or spatial domains under the control of complex perceptual models (as in [18]) to embedding data within frequency domain coefficients (of the well-known transformations, e.g. DFT, DCT, wavelet transformations … etc.), where significance-redundancy properties can be well-defined in terms of the *spectral components* of the host signals, while the robustness of these components to many of the well-known attacks (such as lossy compression, linear and even nonlinear filtering) can be defined in terms of the accurate models that frequency domain provides about these signal processors.

CHAPTER THREE

The Proposed Modified Spread Spectrum Watermarking System

3.1 Introduction:

It has been stated earlier that a better data embedding system is a one that can provide more control over the rate-distortion-robustness tradeoff from the view point of its corresponding application. This actually is an important issue when considering the performance of the different embedding algorithms.

In the next section, some vital modifications to the general spread spectrum watermarking model (discussed in sec. 2.7) will be proposed, those will be proved (later in this chapter) to cause no degradation to system performance, yet, to provide some extra control over the aforementioned tradeoff through the opportunity they offer to *relatively adapt* the embedded data to the local properties of the underlying host signal (which would significantly improve the perceptual fidelity of the system outcome) without inducing any adaptation distortion into the embedded data signal (as will be seen in the next chapter).

3.2 The Proposed Modifications:

Before proceeding with describing the proposed modified spread spectrum watermarking system, a short stop will be made here to examine the proposed modifications in some detail.

It is already known that the robustness of the watermarks encoded using spread spectrum techniques depends -generally- on the dimensionality (length) N of the spreading sequences used to encode them, which actually determines the amount of the overall interference power reduction (relative to the power of the embedded watermarks) that can be achieved -on the average- by these sequences (assuming a fixed size L for message sets). This means that a different spreading sequence (or at least a one with a different length) is required each time a different

robustness level is to be achieved*. Although this is not an inhibitive limitation when using key-dependent spreading sequences generators (rather than dictionaries of spreading sequences), it can still limit the system adaptability in many ways. In order to overcome the aforementioned adaptability problems, it is proposed here that it would be much better if this *rigid* power spreading scheme has been transformed into a more *granular* scheme (a one that consists of many smaller -hence, more manageable- components).

The proposed modifications to increase the granularity of spread spectrum systems here are based on the principles lying behind *integrate-and-dump* receivers used within baseband pulse code modulation (PCM) communication systems (see sec. 11.1 of [29] for more details). An integrate-and-dump receiver integrates (accumulates) the incoming noisy baseband signal over nonoverlapping boundary-synchronized intervals of duration T (the duration of the pulses used to encode the individual PCM messages) before quantizing the output of the integrator sampled (dumped) at the end of each interval to interpret the individual received messages**. This integrate-and-dump preprocessing (or pre-filtering) actually aims at boosting the baseband signal relative to the additive noise inserted through the channel. That is, given some baseband signal that has been disrupted by some *white* additive noise (as is shown in Fig. 3.1), integrating this composite signal over boundary-synchronized intervals of duration T would -separately**- accumulate its value over the individual PCM pulses, such that the accumulated magnitude of the baseband component would linearly increase with duration T, while the standard deviation of the accumulated white additive noise component would increase as slow as $\sqrt{T}$, hence, giving the baseband component a power

* Unless the perceptual fidelity is to be altered, in which case the power of the embedded watermarks rather than the lengths of spreading sequences is modified.

** Note that the integrator should be reset to zero before the start of each new interval.

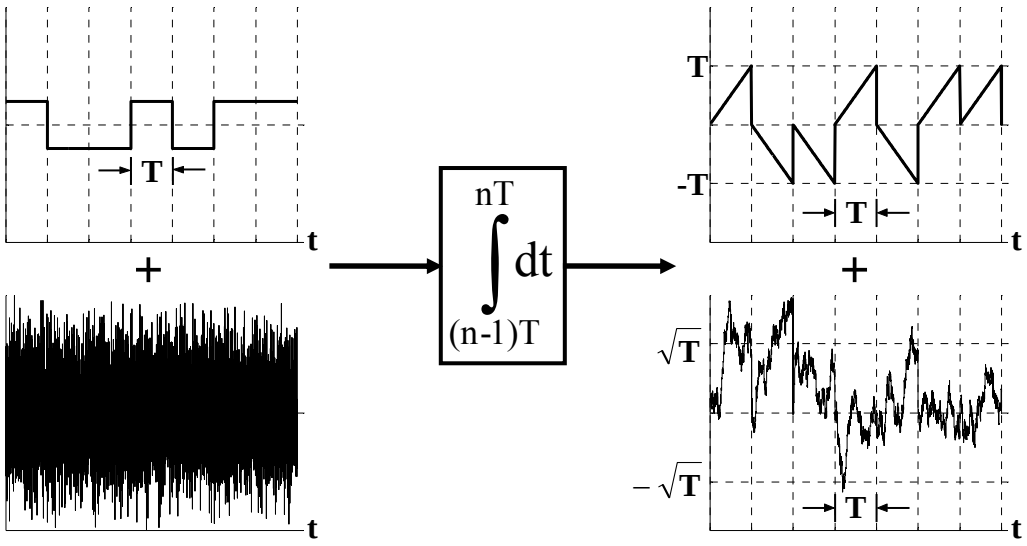


Fig. 3.1 The effect of integrating a baseband signal disrupted by white additive noise (both with normalized power) over boundary-synchronized intervals. Note that integrate-and-dump processing includes sampling the integrator output at the end of each T-duration interval, too.

advantage equal to the duration T over the inserted white noise component.

On the basis of the aforementioned principles, the traditional rigid power spreading mechanism is proposed to be segmented here into a number of *smaller* and *redundant* spreading mechanisms, with the reduction in the processing gain that results from the reduction in the dimensionality of the spreading sequences being compensated for through the employment of the integrate-and-dump technique -discussed above- to take advantage of the available redundant segments in order to achieve their aggregate effect.

Here, each N-dimensional spreading sequence Q^N is to be replaced by an integer number (N/D) of repetitions of some D-dimensional sequence Q_s^D (with D << N for more granularity). This means that each message M is to be repeatedly encoded for N/D times using the same Q_s^D spreading sequence, with the aggregate N-dimensional watermarking array W^N being constructed by tiling these N/D encodings (or watermark segments) W_s^D in a nonoverlapping fashion. At the receiving end, the embedded watermark W^N gets noised by some error signal Z^N that represents the

error induced by both the host signal (or the residual resulting from subtracting some imperfect approximation of the host at the receiver) and the attack channel into the generated watermark estimate array $\hat{W}^N$:

$$z^N = \hat{w}^N - w^N \tag{3.1}$$

The watermark segments estimates $\hat{W}_s^D$ constituting the watermark estimate array $\hat{W}^N$ are then accumulated (or equivalently averaged) into one segment $\tilde{W}_s^D$ with an enhanced signal-to-noise ratio (the ratio of the power of the embedded watermark segments to that of the error induced into these segments) taking advantage of the available redundancy. It is this enhanced segment $\tilde{W}_s^D$ that is then decoded for an estimate of the embedded message $\hat{M}$ using the spreading sequence Q_s^D (or the matched filters with the sequences $\{Q_{sj}^D \mid j = 1 \ldots L\}$ if this has been the corresponding encoding scenario, see sec. 2.7).

The important aspect here is that of making sure that the error signal Z^N inserted into each watermark estimate array $\hat{W}^N$ is such distributed over the D dimensions of the constituting watermark segments $\hat{W}_s^D$, that the error signal inserted into each dimension (over all the segments to be accumulated) is white (see Fig. 3.2). This way the watermark estimate array $\hat{W}^N$ can be viewed as consisting of D parallel baseband channels (corresponding to the D dimensions of the constituting watermark segments over each of which the latter assume a fixed value -as the dashed horizontal line on the magnified signal graph in Fig. 3.2 shows-) each being disrupted with some white noise signal. Now, an average power enhancement equal to the number of accumulated watermark segments N/D (the duration over which accumulation takes place on each dimension) can be achieved, which would *compensate* for the aforementioned reduction with the processing gain.

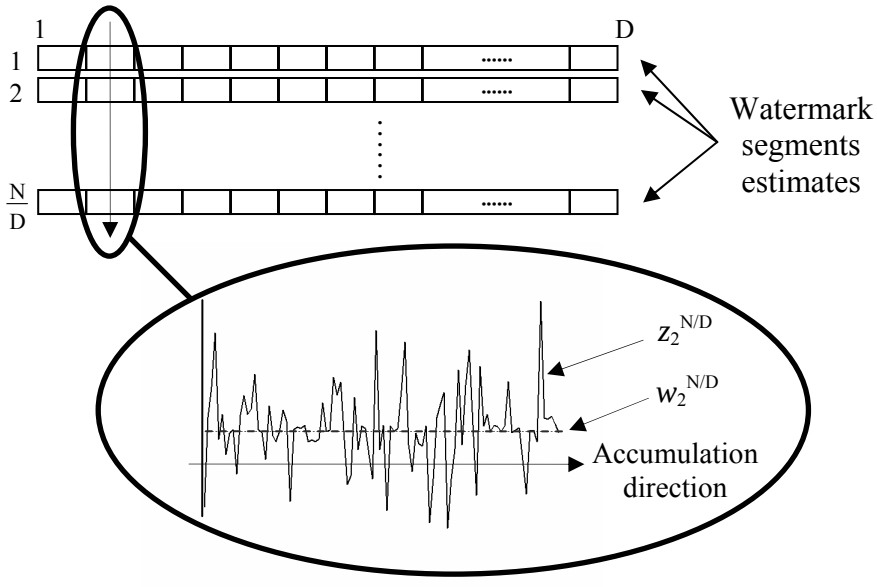


Fig. 3.2 The construction of watermark estimate arrays $\hat{W}^N$.

In the next section, the modifications discussed above will be implemented on a spread spectrum *image* watermarking system, so that it would be possible to practically evaluate their performance later. However, with the appropriate adjustments on the design details those are dependent on the type of data to be marked, the system described next can easily be extended to any other multimedia data type.

3.3 The Proposed Modified Spread Spectrum Image Watermarking System:

The block diagrams of the proposed modified spread spectrum watermark embedding and decoding systems are shown in Figs. 3.3 and 3.5, respectively. However, the common component between these two systems will be discussed first in here, that is:

3.3.1 The Spreading Sequences Generator:

It has been stated in sec. 3.2 that in order to get a more granular spreading mechanism, the traditional spread spectrum watermarking system (discussed in sec. 2.7) needs to be modified, so that much shorter

spreading sequences can be used. The proposed reduction in the lengths of spreading sequences, however, has some disadvantageous impact on the security of the system in terms of immunity to brute force attacks. To overcome these disadvantages, a number of the *uncorrelated* spreading sequences are used here (rather than using a single sequence), in order to increase the time complexity of the former attacks and to reduce the chances of an attacker to make an estimate of the spreading sequence through averaging several marked host blocks (as will be seen later). Figs. 3.3 and 3.5 show that the spreading sequences here are generated through two cascaded steps:

1. A pseudorandom generator is first seeded with the secret key (Key 1) to generate up to D independent and real valued D-dimensional sequences, where D is the length of the spreading sequences segments to be used here to encode the messages.
2. The generated random sequences are then *decorrelated* using the *Gramm-Schmidt orthonormalization procedure*. This procedure can generally be described by the pseudo-instructions below:

   for i := 1 to F

   $$q_i^D = q_i^D - \sum_{j=1}^{i-1} < q_j^D, q_i^D > q_j^D$$

   $$q_i^D = \frac{q_i^D}{\left\| q_i^D \right\|}$$

   end.

   where $F \le D$ is the number of sequences to be orthonormalized and $\left\| q^D \right\| = < q^D , q^D >^{1/2}$ is the norm of the corresponding vector [12].

   Note here that the computational complexity of this procedure is of the order $O(F^2D)$, which makes it very important an issue to choose a suitable number of spreading sequences F, so to achieve both an acceptable execution time and security level, taking into account

that the latter number is upper-bounded by the length of these sequences D, which determines the maximum number of orthogonal vectors those may span the D-dimensional space, (hence, relatively upper-bounding the system security).

The output of the spreading sequences generator consists of a set of D-dimensional orthonormal spreading sequences segments $\{Q_j^D | j = 1 \ldots F, 0 < F \leq D\}$, where $< q_i^D, q_j^D > = 0 \ \forall \ i \neq j$, and $< q_i^D, q_i^D > = 1$.

3.3.2 The Watermark Embedding System:

As is shown in Fig. 3.3, the proposed watermark embedding system receives five necessary inputs. These inputs are: the host image to be marked, the message sequence to be embedded (assuming here that the message consists of -or can be converted into- a binary sequence), the set

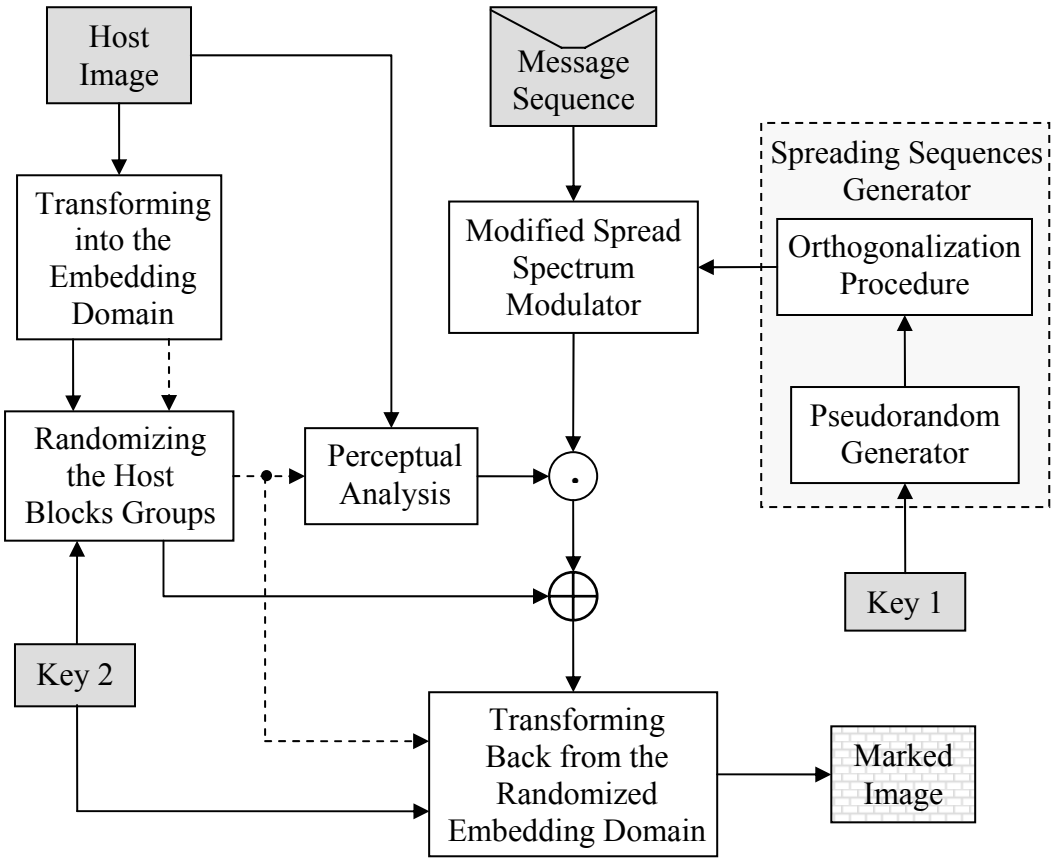


Fig. 3.3 The proposed modified spread spectrum watermark embedding system. Note here that the dashed arrows correspond to the passage of the complete DCT coefficients blocks (rather than blocks of selected low frequency coefficients).

of orthonormal spreading sequences segments produced by the spreading sequences generator, the key (Key 2) necessary for randomizing host blocks groups and the embedding control parameters (not shown in the figure) those determine the ratio between the power of the host signal and that of the embedded watermark (global embedding SNR) and the number of embedding repetitions of each *distinct* watermark segment (the number of times each bit in the message sequence is encoded).

The proposed embedding system involves the following stages:

1. The algorithm starts by transforming the input host image into the domain where watermark embedding is to take place. To achieve high robustness here, the watermark is chosen to be embedded within the low frequency coefficients (excluding DC) of the 2-dimentional DCT of the 8*8 blocks of the image luminance component. This selection actually corresponds to the perceptually most significant components of the image (the least modified -or quantized- components by JPEG compression algorithms and hopefully are also the least modified by many other signal processors). Blocks of selected low frequency coefficients are then passed to the next stage of the embedding system.
2. The passed host coefficients blocks are then grouped together to form larger *groups* of blocks (corresponding to the rooms where watermarking arrays are to be embedded) with certain statistics those are necessary to comply with the assumptions made in sec. 3.2 about the properties of noise signals (inserted into baseband channels) required to optimize the performance of integrate-and-dump receivers. Here, the C-dimensional host blocks (where C is the number of low frequency coefficients within each host block) are grouped into (N/D)-by-(D) coefficients arrays (as is shown in Fig. 3.4), where (N/D) is the number of watermark segments

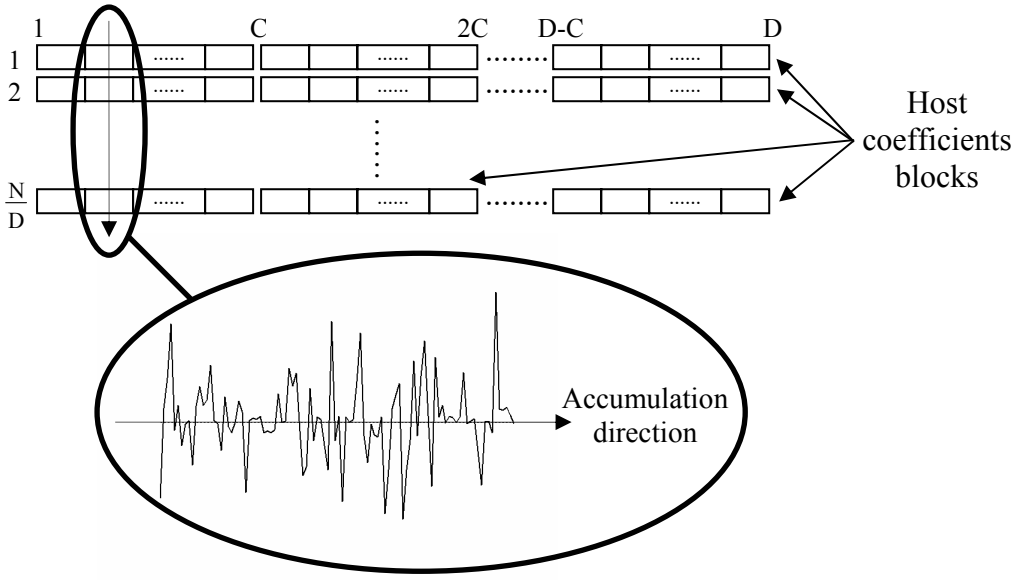


Fig. 3.4 The construction of host blocks groups (coefficients arrays). Note here that each host block contains C smaller boxes representing the coefficients constituting that block. Note also that the coefficients constituting the individual columns of the array are assumed to constitute some white noise signals, as is shown by the magnified signal graph.

embedding repetitions required by the user (shown here as a fraction in order to comply with the symbolism used in sec. 3.2), while D is the dimensionality of these segments. Here, D should be an integer multiple of C, such that watermark segments can lie down over complete host blocks boundaries. This actually is important to ensure that only coefficients with equal frequency indexes are accumulated through the columns of the arrays formed by these blocks, in order to achieve symmetrical distributions* for the coefficients under these accumulations, which would significantly reduce host interference.

What is important about the aforementioned grouping process is the randomization imposed by this stage on the selection of the host blocks within these groups. Actually, this is crucial to provide the necessary security (see about autocorrelation attacks later) and

* Note here that due to the high energy-compaction property of the discrete cosine transformation, the variance of the DCT coefficients of common images rapidly decreases with the increase of frequency indexes of these coefficients. Accordingly, accumulating through a mix of DCT coefficients with different frequency indexes could ruin the symmetry established within the distributions of equally indexed coefficients.

also to distribute the *local stationarity* of image statistics over the different groups to ensure the whiteness of the inserted host interference signal into the accumulated blocks within these groups (see Fig. 3.4). This randomization is generally controlled by the secret key (Key 2).

3. To encode the message bits then, they are used to modulate the corresponding spreading sequences segments within the set of orthonormal sequences supplied to the input of the modified spread spectrum modulator (note here that since any message can be represented by a bit sequence, restricting the modulator input to binary sequences will not involve any loss of generality). To encode a message bit, the modulator starts by selecting a sequence -from the set of spreading sequences- (using a circular selection rule to achieve a uniform distribution over all the sequences*), whose elements are then to be multiplied by the *quantized value* of the specific message bit in order to modulate the selected spreading sequence with that value. Using a bipolar representation for binary digits here (this corresponds to the use of quantized values {-1, 1} to represent binary {0, 1}) could reduce the modulation mechanism into a simple sign change/unchange (multiply by -1 or not) mechanism. The modulator then adjusts the power of the generated watermark segments according to the global power properties of the host signal, in order to achieve the embedding SNR required by the user**. That is, given a global power scale of β, the overall bit modulation process can be mathematically expressed as:

$$w_i^D = \beta \cdot (2m_i - 1) \cdot q_j^D \qquad (3.2)$$

* A random selection rule with a uniform distribution over the sequences in the set can be used here to further increase the system security. However, this would unnecessarily increase the implementation complexity.

** Note that this magnitude scaling process would not interfere with the sign-based message bits modulation.

where j is the index of the selected spreading sequence (according to the circular selection rule), i is the index of the message bit with respect to the message sequence, $(2M_i-1)$ is the bipolar representation of that bit, while W_i^D and Q_j^D are the corresponding D-dimensional watermark segment and spreading sequence segment, respectively. After modulating all the message bits, the resulting encodings (watermark segments) are then to be duplicated for N/D times -each- to construct the corresponding watermarking arrays, in preparation for embedding them within the host blocks groups (host coefficients arrays) previously packed.

4. To improve the perceptual quality of the system output, the generated watermark arrays are then to be adapted to the local properties of the host signal prior to embedding them. An important advantageous property of the proposed system here is its granular implementation of the embedding redundancy, which has enabled the application of some form of host-adaptation that does not actually impose any adaptation distortion into the embedded watermark. Referring to Fig. 3.3, it is noted that the perceptual host-adaptation is carried out in two separate steps: the first -and the most important- constitutes in perceptually analyzing the host signal -according to some perceptual model- in order to find the appropriate scales those would *relatively* increase the imperceptibility of the corresponding watermark components. The actual scaling process, however, takes place in the second step denoted by the dot operation in Fig. 3.3. For the remaining of this chapter, host-adaptation will not be discussed any further. Host-adaptation will be the subject of the next chapter.
5. The adapted watermark arrays are then to be added to the host blocks groups (coefficients arrays). Here, the N/D repetitions of

each distinct watermark segment (distinct message bit encoding) are to be inserted into the N/D rows of some (N/D)-by-(D) host coefficients array (see Fig. 3.4), such that each host array would contain only (and all of the) similar watermark segments (note here that due to the applied adaptation process, the watermark segments corresponding to the same message bit encoding may not be any more *exactly* the same, however, they may still be highly correlated).

6. At the final stage of the embedding procedure, the marked host blocks groups are decomposed back into their constituting coefficients blocks those are to be substituted into their corresponding low frequency rooms within the complete DCT host blocks (passed from the second stage of the embedding system). The randomization process performed on the host blocks is then to be reversed before applying the blockwise inverse DCT to produce the final marked image.

3.3.3 The Watermark Decoding System:

As is shown in Fig. 3.5, the proposed watermark decoding system receives four necessary inputs, these are: the marked -possibly corrupted- image, the set of orthonormal spreading sequences segments produced by the spreading sequences generator, the randomization control key (Key 2) and the control parameters corresponding to those were used in the embedding system previously described (not shown here in the figure). The proposed decoding system consists of three main stages, the first two of which have been described in detail in the context of the watermark embedding system and are not to be discussed any further in here. However, the proposed modified spread spectrum demodulation stage will be discussed next.

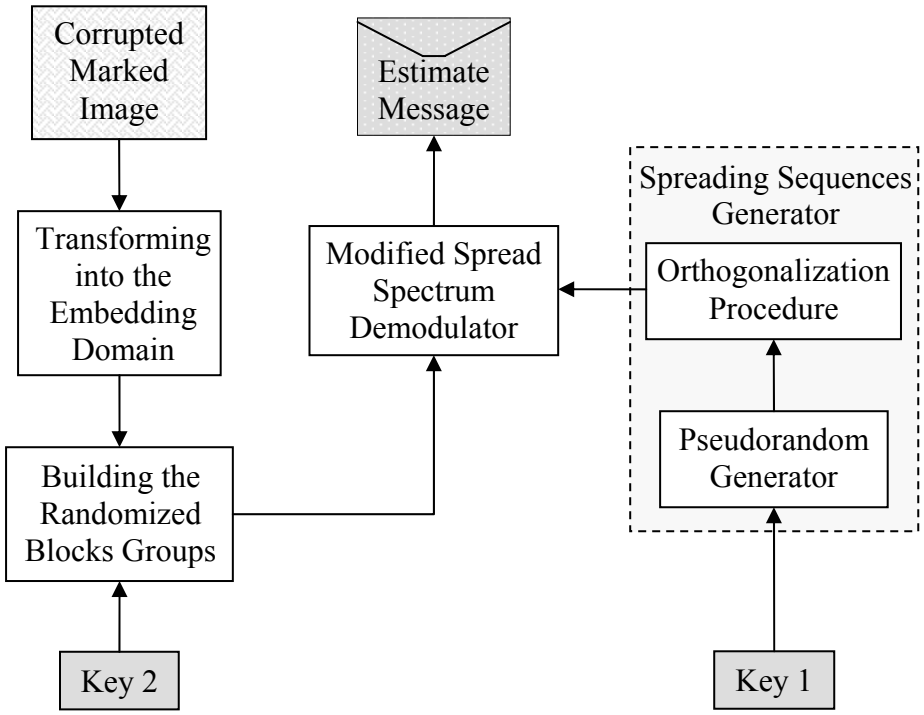


Fig. 3.5 The proposed modified spread spectrum watermark decoding system.

At the receipt of the marked image coefficients arrays (marked blocks groups), the demodulator *sequentially* processes these arrays to build an estimate of the embedded bit sequence. Note that due to the inability of the decoding system to make a useful approximation of the host signal here, the received image coefficients arrays themselves are to be treated as an estimate of the embedded data (a general blind decoding scenario*). Initially, the rows of each array are accumulated (or equivalently averaged) into one row vector that represents an *enhanced estimate* of the distinct watermark segment (bit encoding) embedded within the corresponding array (note that the latter enhancement of the extracted watermark segment with respect to the host image and channel induced interference -or error- is determined by the embedding repetition parameter N/D input by the user). To decode the embedded message bit then, the demodulator here selects the spreading sequence corresponding

* It should be emphasized here though that the availability of any side information (about the host signal) to the decoder can still be utilized to enhance the generated estimate of the embedded data as has been described in sec 2.7.

to that been used by the modulator (note that a synchronized decoder should always be able to select the correct sequence out of the set of spreading sequences), which is then to be correlated with the generated enhanced watermark segment estimate to recover the *modulating* quantized value of the message, whose sign would give the best possible [29] estimate of the bipolar representation of the embedded bit:

$$\hat{m}_i^{bip} = \text{sign}(< \tilde{w}_i^D, q_j^D >) \tag{3.3}$$

where $\hat{M}_i^{bip}$ is the bipolar representation of the ith message bit, $\tilde{W}_i^D$ is the enhanced watermark segment estimate, while i, j and Q_j^D being defined as in equation (3.2). Note that the sign function here discards the magnitude information contained within the result of the correlation process, the fact that has enabled the global power-based magnitude scaling process on the encoder side.

The overall processing gain of the proposed watermarking system can mathematically be expressed in terms of the product of the accumulation interval gain (N/D) by the spreading dimensionality gain (D):

$$G_p \propto \left(\frac{N}{D}\right) \cdot D = N \tag{3.4}$$

Expression (3.4) clarifies the relationship between traditional spread spectrum watermarking systems and the proposed modified system. Note here that according to this expression, the proposed system approaches pure spread spectrum systems as the dimensionality of the watermark segments approaches that of the whole watermarking arrays (hence, the repetition factor $N/D \rightarrow 1$ and accumulation processing approaches zero), while on the other extreme, reducing the length of watermark segments up to unity (when D = 1) would set this system to work as a pure integrate-and-dump receiver with accumulation interval (N/D = N), hence, making the system *fully granular*, but also much less secure

(remember though that in order to comply with the assumption made in stage 2 of the proposed embedding system -discussed in sec. 3.3.2- regarding the need for the distributions of the coefficients under accumulation to be symmetrical, it turns out that the length of the watermark segments cannot be less than C -the number of frequency coefficients within each host block-).

An important issue to note here is that unlike the case of the generic spread spectrum watermark decoding model (shown in Fig. 2.8), in the proposed decoding model here, there is no need for any special processing to equalize the effect of the adaptation process on the embedded data signals. This aspect will be explored in the next chapter.

3.4 Experimental Results:

At the last part of this chapter, the robustness of the proposed watermarking system is to be experimentally evaluated against a wide range of the well-known attacks.

3.4.1 Selection of Test Images:

To ensure the generality of the experimental results to be gathered later, the tests here are to be applied on a number of images with different textures and with different levels of details, brightness, and contrast. The photos shown in Fig. 3.6 may well satisfy these requirements, and are hence to be used through all the tests to be performed throughout this thesis. These images are all (600*448) pixels sized and are all 8-bit grayscale bitmaps. Note that although other color images could have been used here instead (remember that watermarks would have still been embedded within the luminance components of these images -to preserve robustness-), grayscale images are seen to better emphasize the visual artifacts imposed by the embedding process, and are hence used in here.

(a) The **bug** test image.

(b) The **buildings** test image.

Fig. 3.6 (a),(b) The selected test images. Note that these images have been scaled into (92% * 92%) of their actual sizes to fit them in here.

(c) The **elephant** test image.

(d) The **vase** test image.

Fig. 3.6 (continued) (c),(d) The selected test images.

(e) The **zebra** test image.

Fig. 3.6 (continued) (e) The selected test images.

3.4.2 Selection of the Appropriate Coefficients for Embedding:

It has been mentioned earlier that in order to achieve high embedding robustness, watermarks here are chosen to be embedded within the low frequency coefficients (excluding DC) of the 2-dimentional DCT of the 8*8 blocks of the image luminance component. Note here that although DC coefficients do correspond to the most significant components of the image (hence, are supposed to be capable of achieving a high robustness level), they are still an unavailable option for embedding. This actually due to the undesirable high correlation that exists between these coefficients and the corresponding image contents, which biases the statistics of these coefficients over the image space in an unmanageable manner (in this case, these undesirable properties couldn't have been overcome by randomization process), hence, making these coefficients a

source of an unacceptable level of interference. Note also that as these coefficients tend to control the average luminance of the corresponding image blocks, manipulating their values may induce some undesirable visible artifacts.

Next, the issue of choosing the best possible set of low frequency (AC) coefficients to maximize the overall embedding robustness is to be considered. However, as it is practically impossible to determine the response of all candidate AC coefficients to all possible channels, the benchmark here is kept as simple as is possible. Figs. 3.7 and 3.8 show the error induced by compression and blurring channels (for different quality factors and window sizes, respectively) into the different DCT coefficients with respect to the variance of these coefficients (note that the zeroth coefficient here corresponds to the DC term).

It can be noticed from Figs. 3.7 and 3.8 that the 20 lowest frequency AC coefficients of the selected test images show an acceptable level of immunity to these channels (remember here that although DC coefficient shows the highest *relative* immunity to these attacks, it still acts as a source of much higher interference power than the other coefficients). For the remaining of this work, embedding processes will be restricted to these selected coefficients.

Fig. 3.9 shows the achievable host interference power reduction ratio as a function of the number of accumulated host coefficients blocks (in analogy to the group size -in units of blocks-) for the different test images. Note that although image data can exhibit some undesirable (from the perspective of integrate-and-dump processing) local stationarity properties those due to the inherent nature of this data type, a good randomization procedure can still reduce the effect of these undesirable statistics. As is shown here, the graphs of power reduction ratios are well linear, and the power reduction ratio here approaches the number of

Fig. 3.7

The Error Iduced by a JPEG Compression Channel into DCT Coefficients

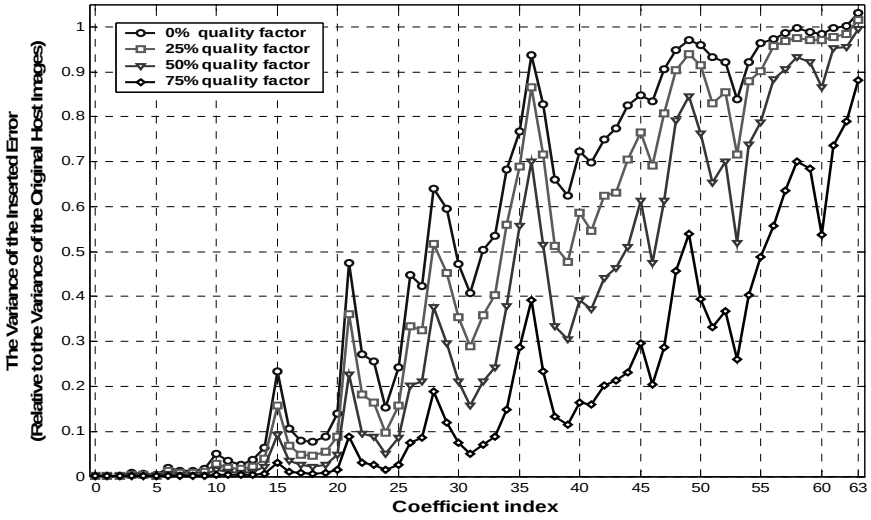


Fig. 3.8

The Error Iduced by a Gaussian Blurring Channel into DCT Coefficients

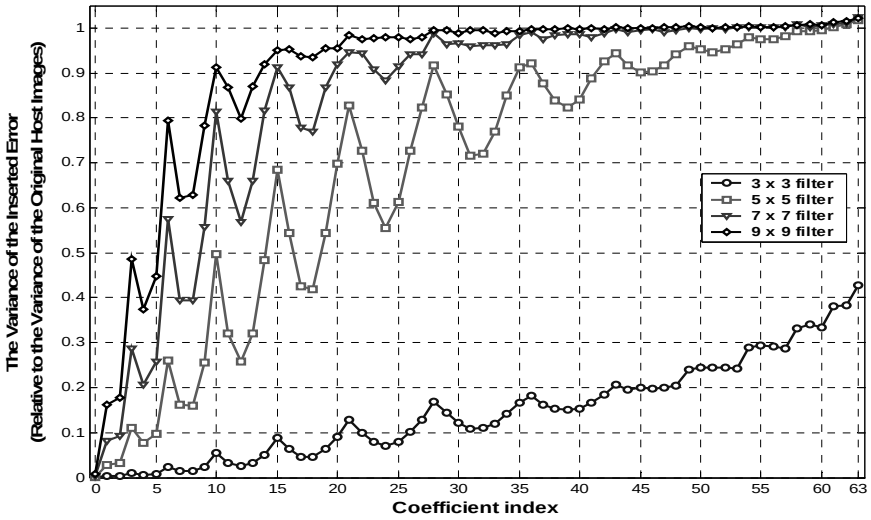
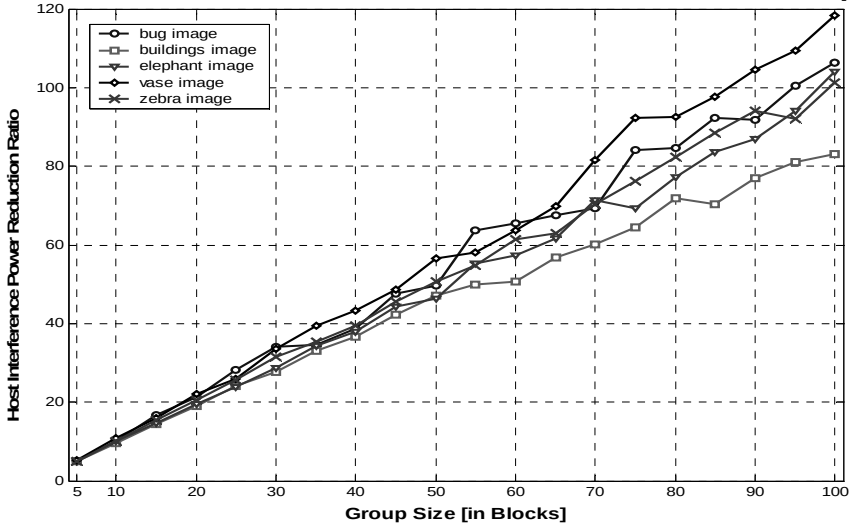


Fig. 3.9

Reduction in Host Interference Power as a Function of Group Size



blocks (of selected coefficients) over which accumulation takes place (which proves the previous expectations about the performance of the integrate-and-dump processing on image data).

3.4.3 Simulation Results:

In this section, the robustness of the proposed system will be experimentally tested against a wide variety of the well-known attacking channels. To ensure an acceptable deal of generality for the results here, the proposed system is tested over a wide range of the different settings for the following parameters*: the embedding repetition rate (the number of times each distinct watermark segment is embedded), which controls the processing gain, the embedding signal-to-noise ratio SNR (or the host-to-watermark power ratio), which -roughly- determines the perceptual quality of the marked image -given a mean squared error measure- and the attack channel parameters (such as the compression quality factor and the blurring filter window size, … etc.), which determines the severity of the corresponding attack and its effect on both the embedded watermark and the host image. However, in order to be able to explore the benefits of the proposed modifications more closely, the lengths of the spreading sequences segments (hence, of the watermark segments, too) used in here will be restricted to the minimum possible (to the length of the host coefficients blocks, which means that the embedding repetition rate here will always be equal to the size of the blocks *groups* -defined in sec. 3.3.2- in units of host coefficients blocks), so to drive the proposed watermarking scheme to its furthest extreme away from traditional spread spectrum watermarking schemes.

Differently sized message sequences (of alternating zeros and ones) are to be used for embedding here, with the decoding bit error rate (BER)

* These correspond to the embedding control parameters (input by user) and the attack channel parameters, (see sec. 3.3.2).

being considered as an indication of the embedding robustness. However, the consideration of the embedding induced visual artifacts is to be delayed to the next chapter.

Results of the tests performed here are shown in Figs. 3.10 - 3.21.

- ***Ideal channel:*** The experimental tests here will be started with the ideal situation, when no attack (modification) is assumed to take place on the marked image before being received by decoder. This scenario actually corresponds to the lower bound on the bit error rates those can be achieved by this system given the corresponding test images and embedding settings. The results of the performed tests on the different test images are shown in Fig. 3.10.
- ***Autocorrelation (mark estimation and removal) attack:*** Before starting to explore the robustness of the proposed system against common signal processors, the immunity of this system to analytical attacks will be first examined. It has been mentioned earlier that for the proposed embedding system to be acceptably secure, then the groups of blocks -of the host image- where similar watermark segments are to be embedded have to *randomly* span the host image space. This shuffling process is actually aimed at preventing the attackers from identifying these similarly marked blocks, averaging them and then subtracting the resultant estimations (which can be proved to be equivalent to those would be generated by authorized decoders) from the corresponding marked image blocks to *remove any trace* (at least from the view point of blind decoders) of this embedded data from the marked image. However, assuming that the redundancy form imposed by the proposed encoding scheme may still be utilized to attack the marked images, the question here becomes whether an attacker -who is unaware of the randomization map used- can cross-

correlate the marked image blocks searching for those whom are carrying similar marks. Here, the best possible (remembering that the correlation coefficients calculated in here are highly affected by the high energy host interference and also by the high correlation already inherent in the adjacent blocks of the image, which may -after all- render this attack useless) autocorrelative attack strategy consists in correlating all the marked image blocks with each other and then *clustering* them (according to their correlation coefficients) into equally sized maximum similarity (correlation) clusters (groups). Note that this problem is clearly *intractable* (computationally difficult) for a sufficiently large number of blocks, which is generally an unavoidable property of image (or any other multimedia) data. However, to give an idea about the possible results of this attack, a *nearest neighbor* approximation [30] have been adopted in here. That is, starting with some arbitrary block, the corresponding number of blocks those are maximally correlated to this block are searched for in order to estimate the first group (cluster) of blocks, then, excluding the blocks just selected from the next searches, these steps are repeated until the last group is identified. The effect of this attack channel on the decoding bit error rate is shown in Fig. 3.11.

- ***JPEG* (*lossy compression*) *channel:*** The first lossy channel (as opposed to ideal channels, whose effect imposes no loss on the information content of the embedded data -from the view point of the corresponding decoders-) to be considered in here is the JPEG lossy image compression channel. This channel can be briefly described as a *weighted* quantization process for the blockwise 2-dimensional DCT coefficients of the image, with the weight (severity) of the quantization process applied to each coefficient

being globally controlled by the required quality factor and the frequency index of that coefficient. The effect of this attack channel* on the decoding bit error rate (given the quality factors: 60%, 30%, 0%) is shown in Fig. 3.12.

- ***Linear/nonlinear blurring channels:*** Three types of image low pass filtering kernels have been used in this test, these are: the Gaussian and uniform average kernels, and the nonlinear median kernel. These filters generally tend to diminish the high frequency contents of the input images (such as objects edges and fine textures) through substituting each pixel with the average of the neighboring pixels within a specified window (using weighted versions of the pixels within that window) or simply replacing them by the median pixel value within the specified window, hence, making the filtered images look more *blurry*. The effect of this attack channel on the decoding bit error rate is shown in Fig. 3.13. Here, 3*3 windows have been used for the average and median filter kernels, while both 3*3 and 5*5 windows have been used for the Gaussian filter kernel (with variances 0.5 and 1, respectively).
- ***Dithering channel:*** In image processing domain, *dithering* generally refers to the process of *artificially* increasing the color contents of the images by varying (controlling) the colors of the adjacent pixels -throughout these images- in such a way to simulate the other unsupported ones (this actually is similar to color mixing). In this test, the color depths (in terms of gray shades) of the marked images have been reduced from 256 different gray shades into only 2 different shades, with the resulting images being

* Note here that the marked images in this tests have been compressed with the aid of the ACDSee 6.0 photo editor JPEG compression utility.

dithered to increase their color proximity to the original grayscale images. The effect of this channel* on the decoding bit error rate is shown in Fig. 3.14.

- ***Histogram equalization channel:*** Histogram equalization is a special *grayscale transformation* that is aimed at improving the overall viewability of images [31]. To do so, this procedure remaps the different luminance levels (grayscales) of the image -to be equalized- using a mapping function that takes the form of the integral of the histogram of the image grayscales, such that the contrast among the different scales is varied according to the number of occurrences of each scale, making their histogram more flat. However, to enhance the results achieved by this processing, the original histogram needs to be binned into a smaller number of scales, which would implicitly decrease the color depth of the equalized image (in terms of gray shades again). The effect of this channel* on the decoding bit error rate is shown in Fig. 3.15.
- ***Cropping, scaling and rotation attacks:*** These attacks can generally be classified as geometrical (desynchronization) attacks. Although** spread spectrum watermarking techniques (generally all the techniques previously described) are essentially not expected to survive these attacks by themselves, the proposed system is decided here to be tested against some slight geometrical distortions. The effect of these channels*** on the decoding bit error rate is shown in Figs. 3.16 through 3.18.

* Note that the dithering and histogram equalization processes here have been performed using Matlab's corresponding functions.

** It is important to mention here that there has been recently a major interest in developing some geometrically invariant watermark mapping techniques those are specifically designed to support the abilities of traditional watermarking techniques to counter local and/or global geometrical distortion attacks.

*** Note that the image scaling process has been handled here with the aid of the ACDSee 6.0 photo editor resizing utility, while the rotation process has been handled using Matlab's image rotation instruction (with bilinear interpolation technique).

Note that although no special *registration* processing has been applied to restore the geometrically distorted images before being decoded in here, there have been still some necessary modifications those needed to be applied to these distorted images before they could have been decoded. This processing includes: *first*, substituting the areas those have been cropped out of the marked images with some equivalently sized blank areas to restore the original dimensions of these images (assuming that the decoding side here is able to identify the location of the cropped part of the image with respect to the original one), *second*, scaling the shrunk images back to their original dimensions (assuming the availability of the original size information to the decoder), *while* for the case of rotation distortions, the extra edges of the rotated images have been cut out to preserve their dimensions (based on the previous assumption, too) with those parts of the images included within these cut edges being discarded.

- ***Collusion attack:*** Here, the attacker (or the colluding party) is generally assumed to have in hand a number of *differently marked versions* of the same host image. So, in order to diminish the watermarking signal embedded in each of them, the attacker here averages the available marked images into a new one (or equivalently generate a new one that consists of a complete set of parts those are gathered from these different images), which would -on the average- contain an m-times weaker version of each one of the original embedded watermarks, where m here is the number of the averaged versions. The effect of this channel on the decoding bit error rate is shown in Fig. 3.19.
- ***Re-watermarking attack:*** The last channel to be considered in here corresponds to the re-watermarking scenario. Here, the attacker

(who is assumed to be aware of the details of the marking algorithm -but certainly not of the specific watermark that has been embedded by the encoder-) passes the marked image -to be attacked- again to the marking algorithm along with his own watermark (or watermarks) in order to: *first*, introduce an additional source of interference (noise) to the embedded mark (this actually corresponds to the best possible noise insertion strategy, as it would synchronize the effect of the inserted noise signal to the original embedded watermark -this is especially correct for the case of host-adaptive marking considered in the next chapter-) and *second*, re-mark this image as his own (note here that the additive nature of the proposed embedding process implies that unless they are especially encoded -to carry some timestamp for example-, none of the embedded watermarks -by both encoder and attacker- can be proved to have been embedded first). However, in the context of the tests in here, the attention will be restricted to the effect of the embedded watermarks on each other in terms of the watermark-to-noise ratio WNR (the ratio between the power of the specific mark to be *decoded* and the power of the other inserted marks). The effect of this channel on the decoding bit error rate is shown in Fig. 3.20. Note here that each re-watermarking process applied by the attacker can also be viewed as an embedding capacity multiplication (in case that the encoder himself has embedded multiple watermarks within the host image).

The effect of each lossy attack channel above on the system performance is summarized in Fig. 3.21 (note that the results shown in there correspond to the bug image, marked at an embedding SNR of 30 dB).

3.4.4 Integrate-and-dump to Spread Spectrum Encoding Tradeoff:

In this section the robustness of the proposed system will be tested over different regions of the integrate-and-dump to spread spectrum encoding tradeoff. This actually corresponds to the tradeoff imposed by the *shape* of the total N-dimensional watermarking array (shown in Fig. 3.2), which specifies the ratio between the number of repetitions of the distinct watermark segment constituting this array (or simply the number of rows N/D of the array that corresponds to the accumulation *interval* of the integrate-and-dump encoding section) and the length of these segments (or equivalently the *dimensionality* of the spread spectrum encoding section that is equal to the number of columns D). Here, the proposed system will be tested with respect to a number of the different spreading dimensionalities (rather than restricting the experiments to the minimum possible spreading dimensionality) to examine the effect of the aforementioned tradeoff on the robustness of the embedding process.

The first test to be performed in here considers the performance of the proposed system over ideal (lossless) channels. A number of the different spreading dimensionalities have been tested here, however, -to be brief- Fig. 3.22 shows only those results corresponding to the spreading dimensionalities 1 block and 10 blocks (remember that each host block is actually 20 coefficients size). The performance of the proposed system under additive Gaussian noise insertion channels conditions is also considered here. Fig. 3.23 shows the effect of this channel over a number of the different watermark-to-noise ratios (WNRs). It should be noticed from Figs. 3.22 and 3.23 that the integrate-and-dump to spread spectrum encoding tradeoff -under consideration- *has no effect* on the system robustness (as it has -theoretically- been proved).

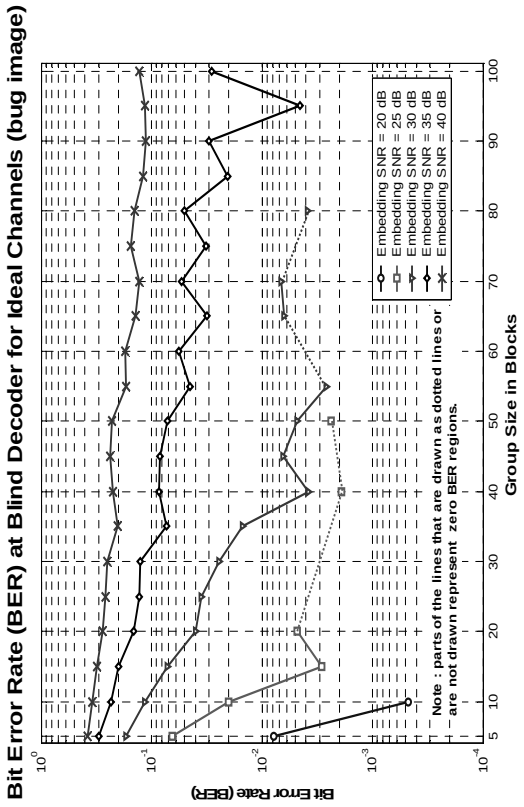


Fig. 3.10 (a)

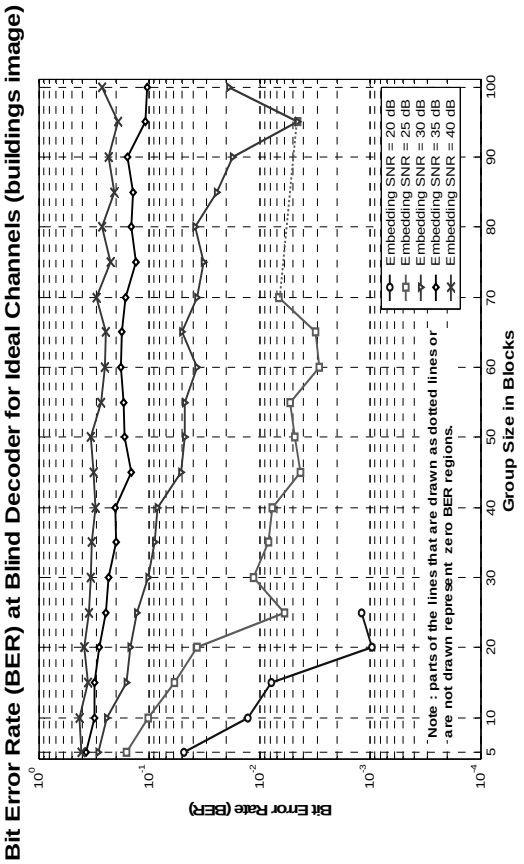


Fig. 3.10 (b)

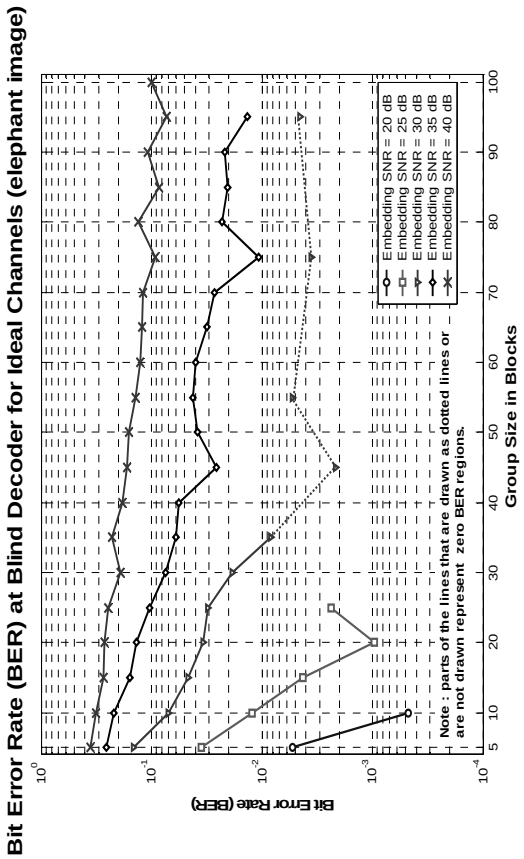


Fig. 3.10 (c)

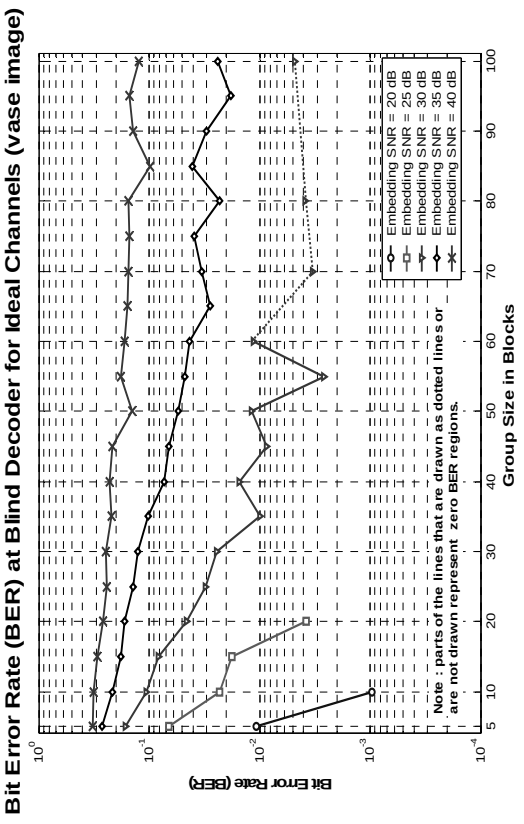


Fig. 3.10 (d)

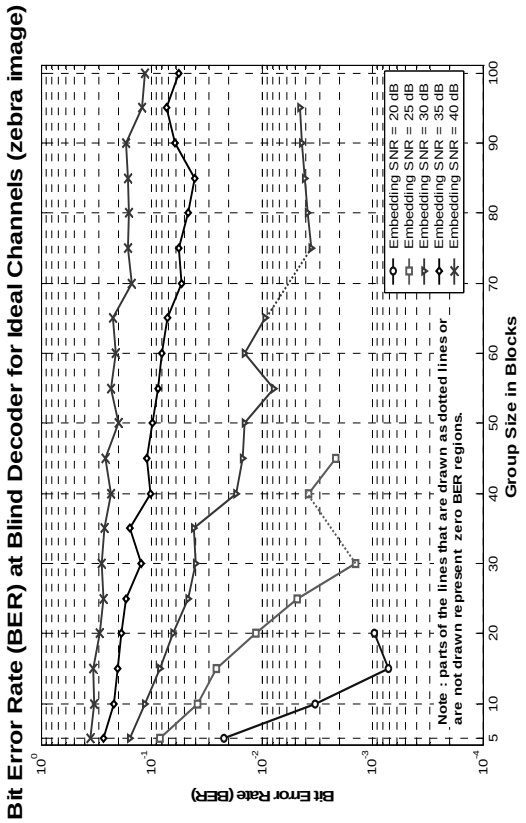


Fig. 3.10 (e)

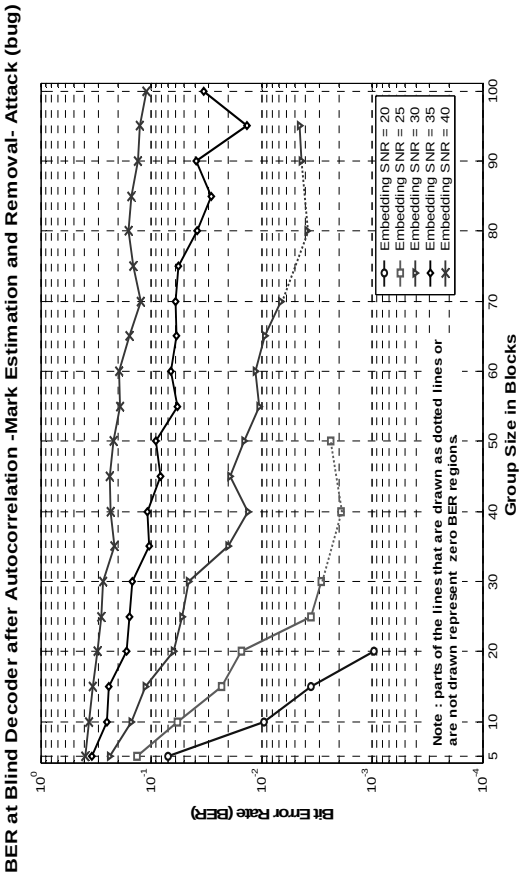


Fig. 3.11 (a)

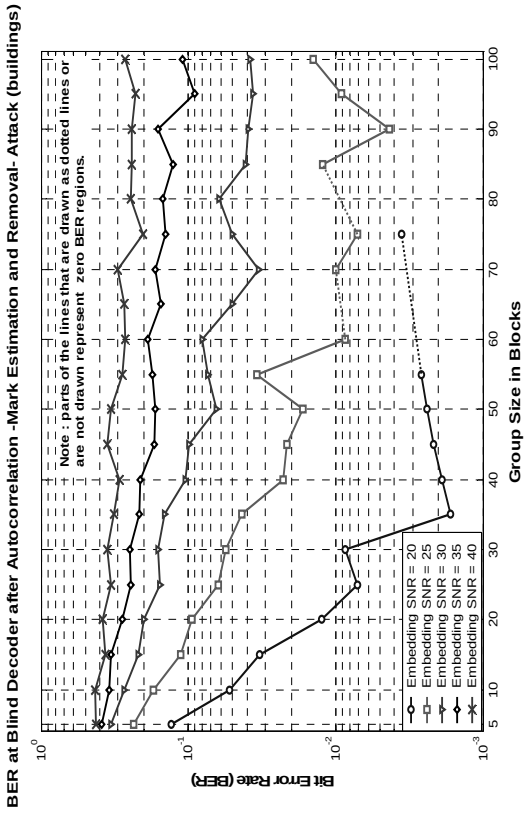


Fig. 3.11 (b)

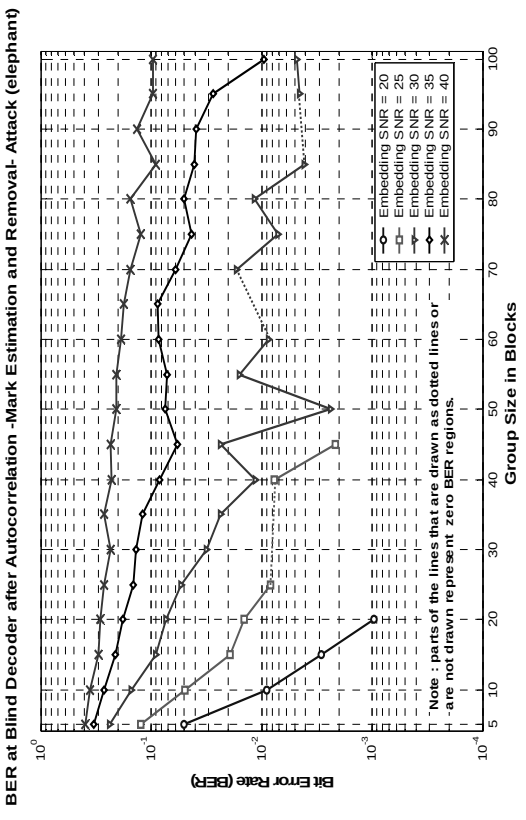


Fig. 3.11 (c)

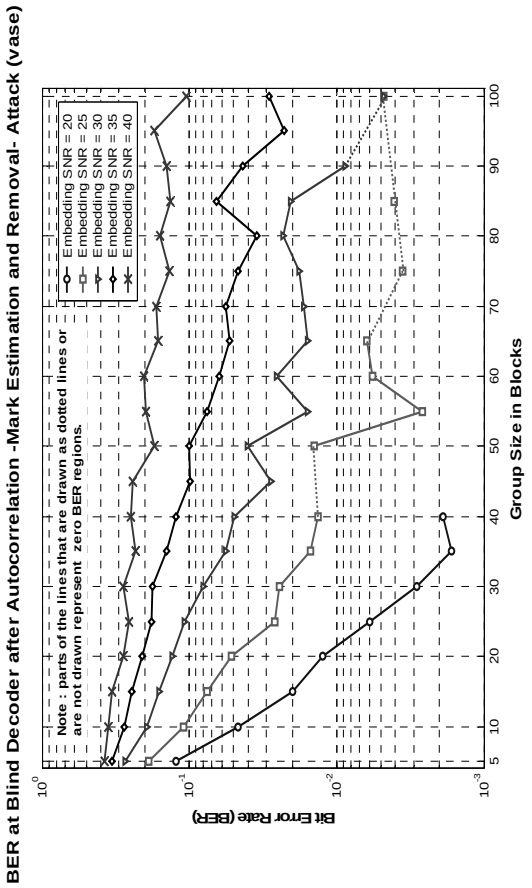


Fig. 3.11 (d)

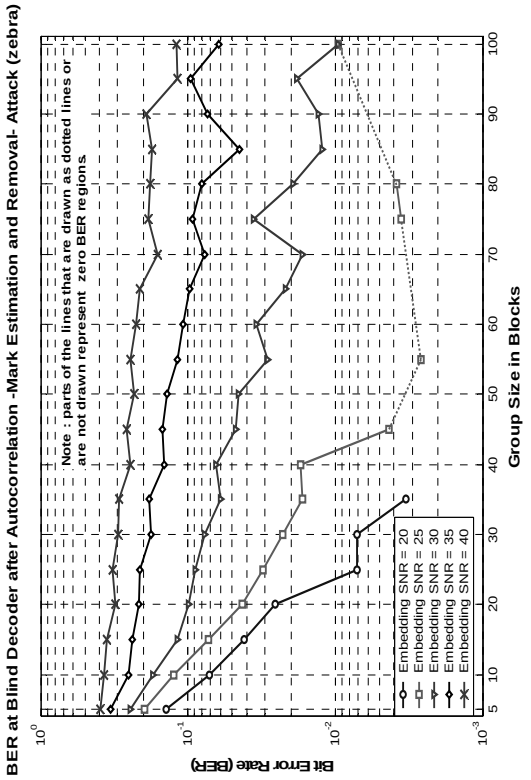


Fig. 3.11 (e)

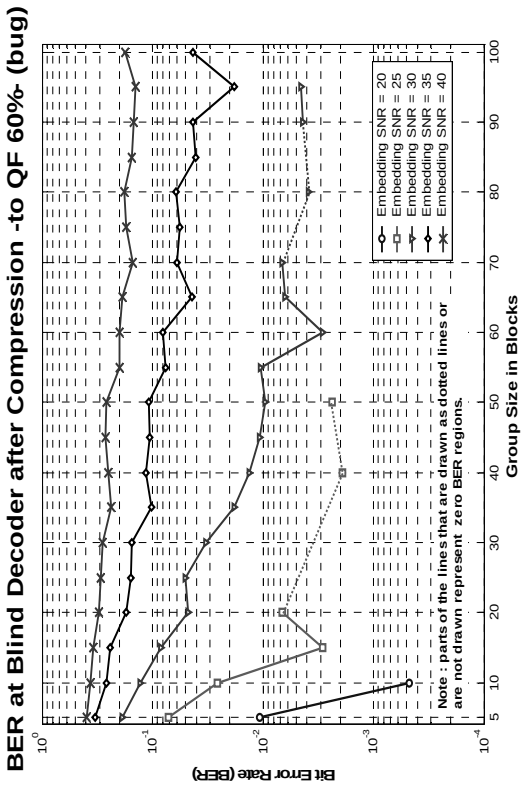


Fig. 3.12 (a)

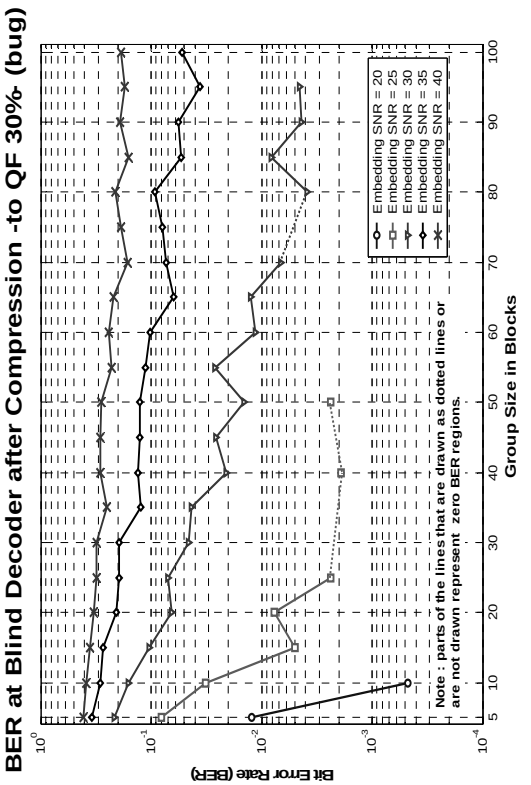


Fig. 3.12 (b)

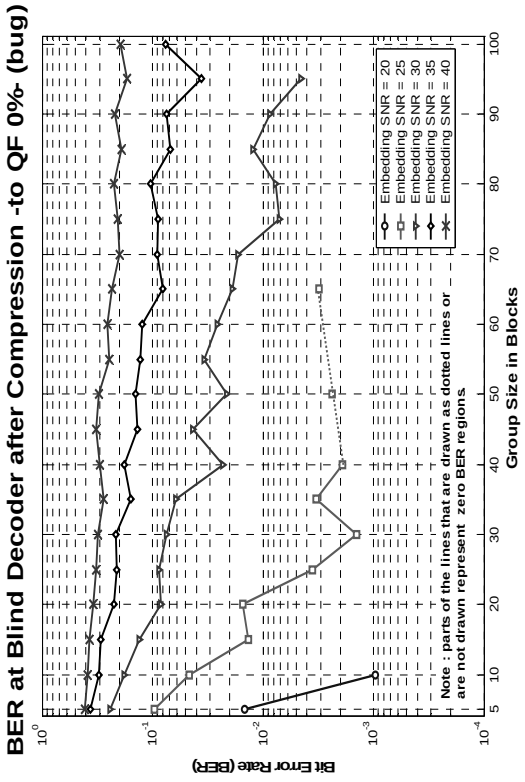


Fig. 3.12 (c)

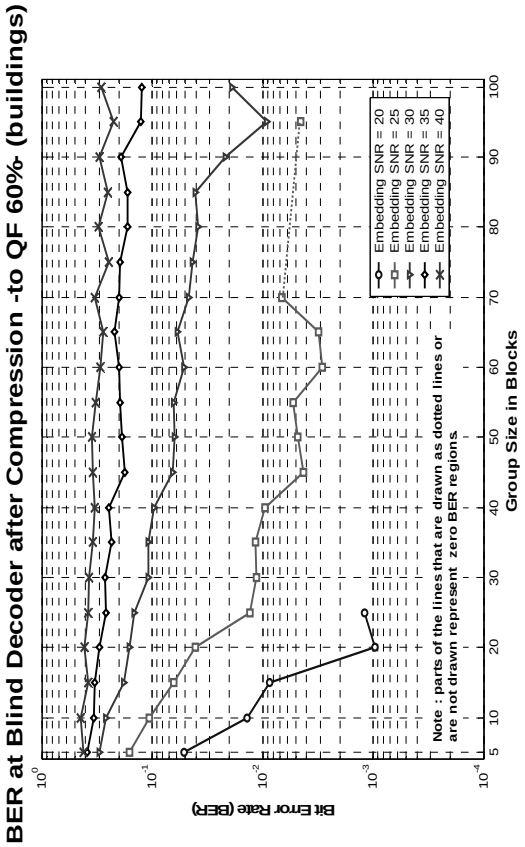


Fig. 3.12 (d)

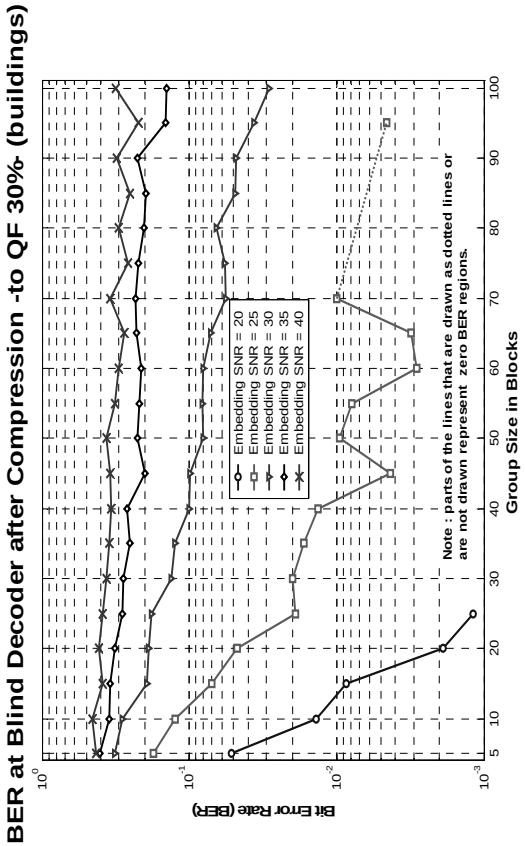


Fig. 3.12 (e)

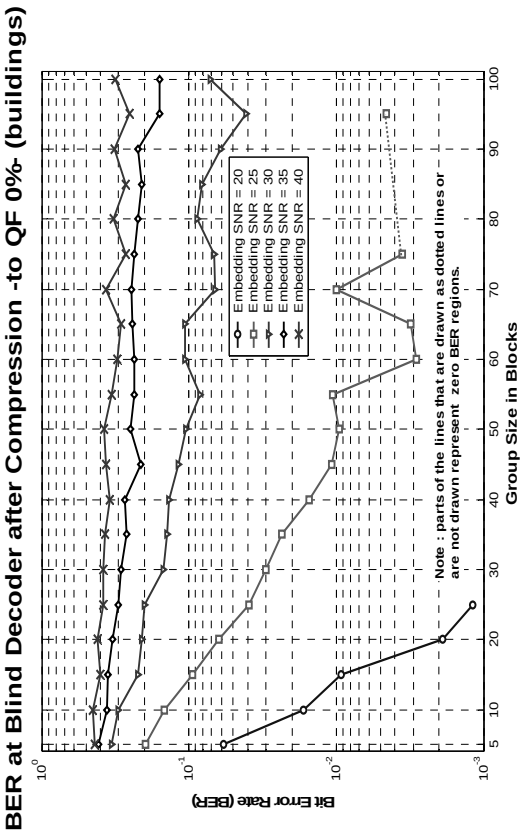


Fig. 3.12 (f)

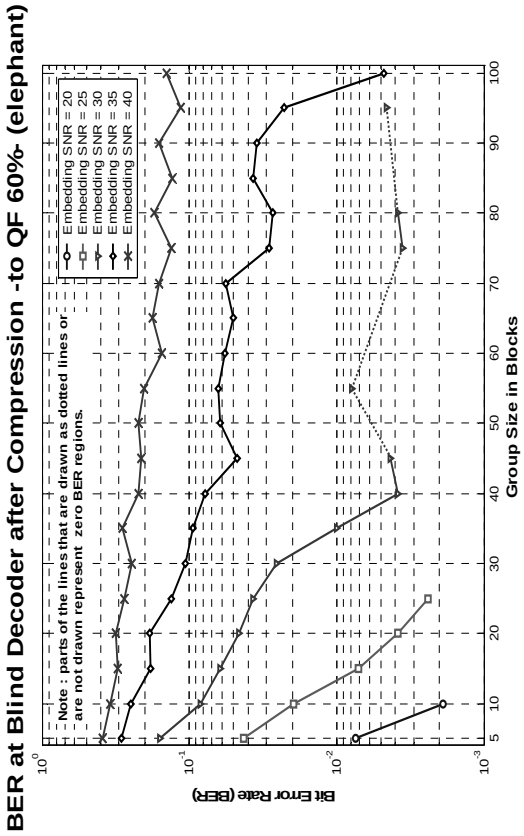


Fig. 3.12 (g)

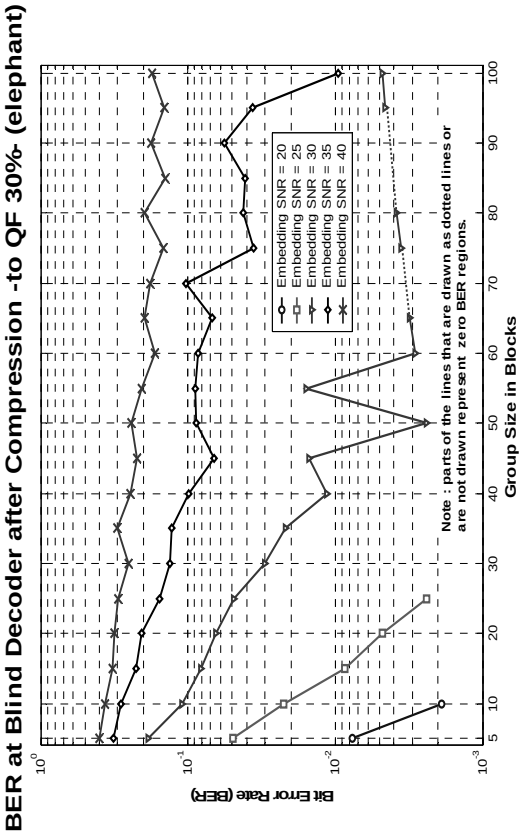


Fig. 3.12 (h)

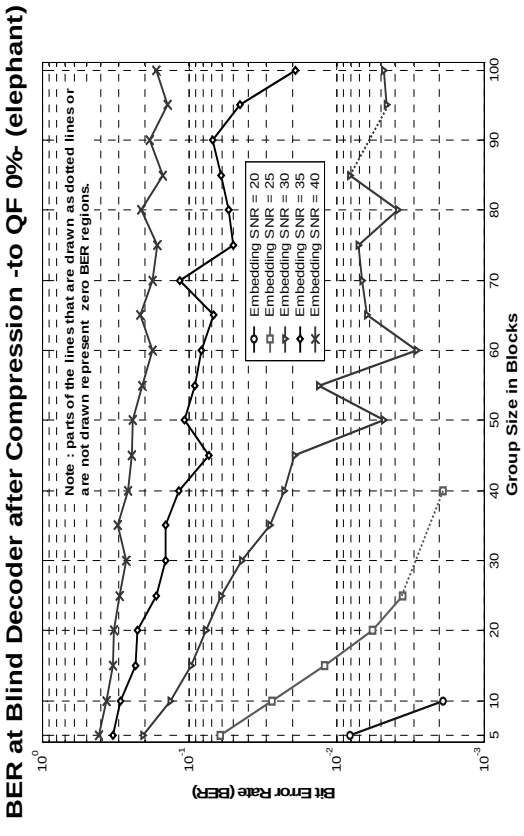


Fig. 3.12 (i)

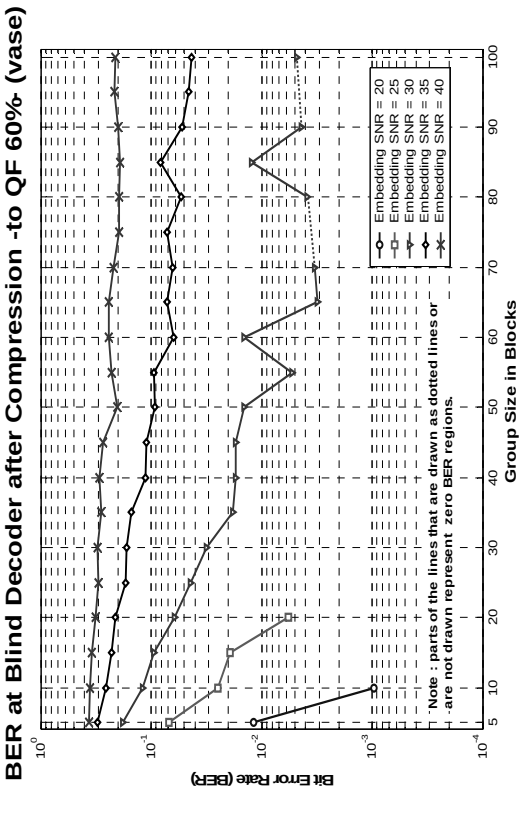


Fig. 3.12 (j)

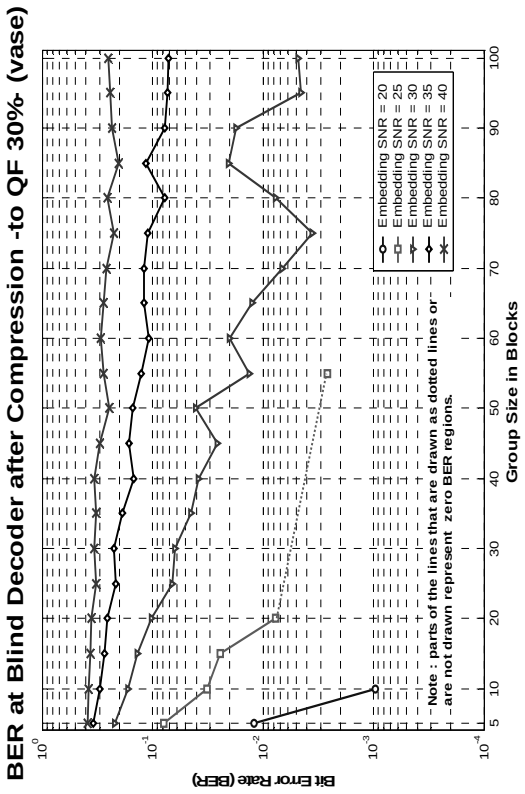


Fig. 3.12 (k)

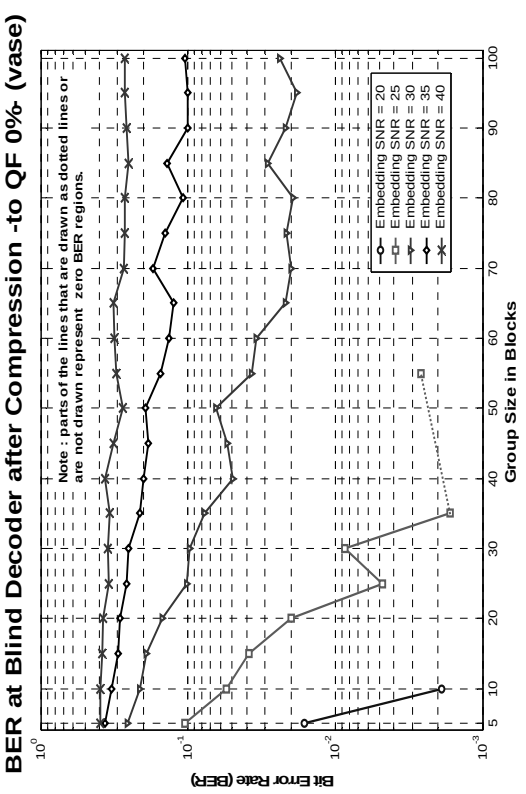


Fig. 3.12 (l)

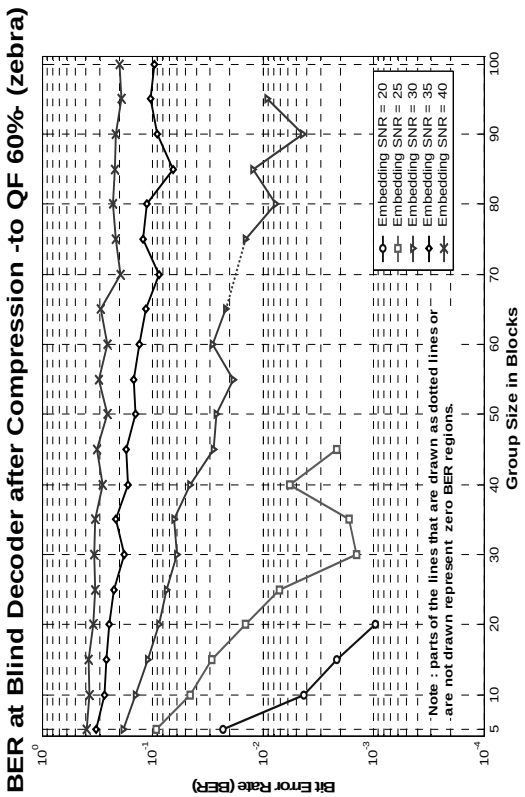


Fig. 3.12 (m)

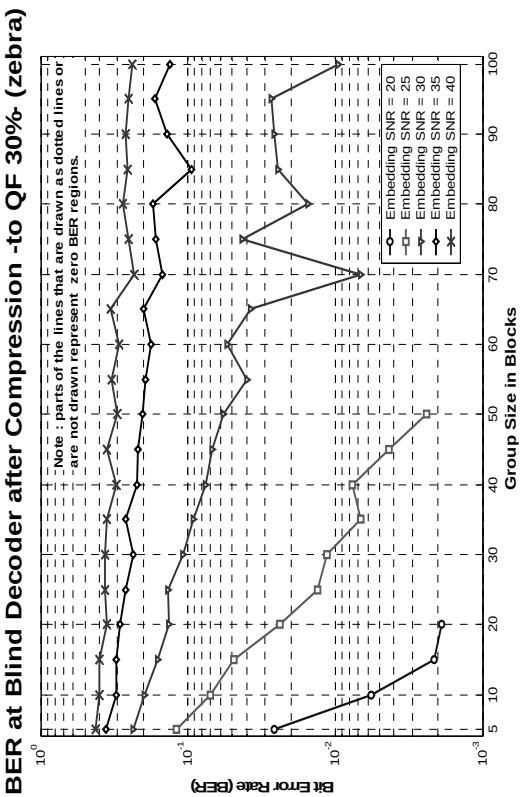


Fig. 3.12 (n)

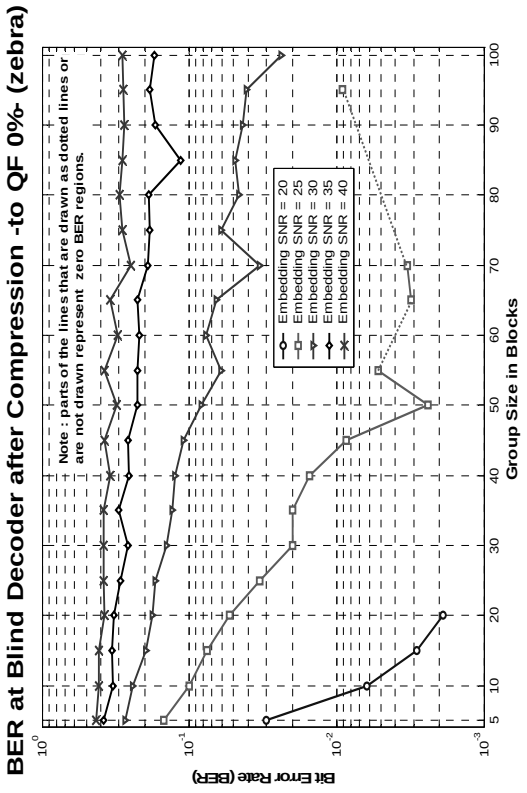


Fig. 3.12 (o)

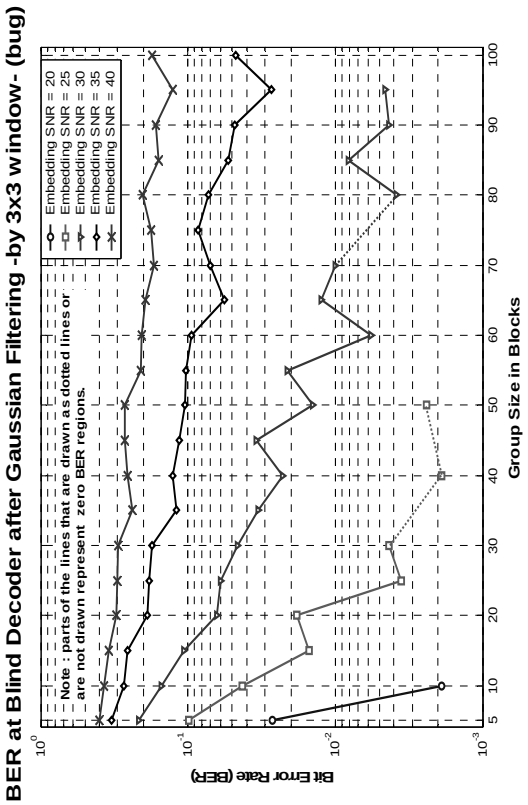


Fig. 3.13 (a)

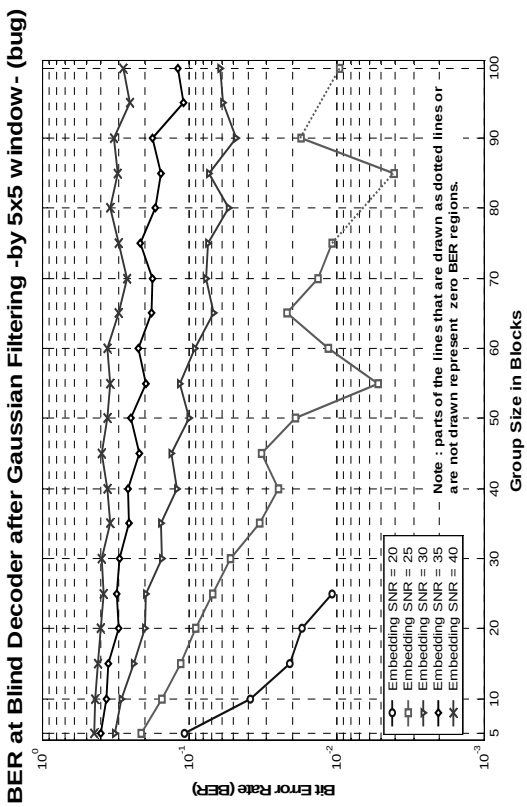


Fig. 3.13 (b)

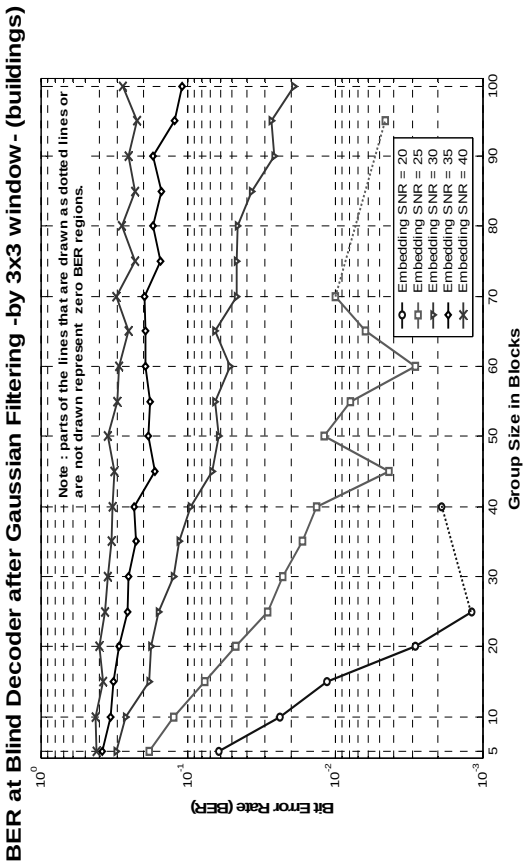


Fig. 3.13 (c)

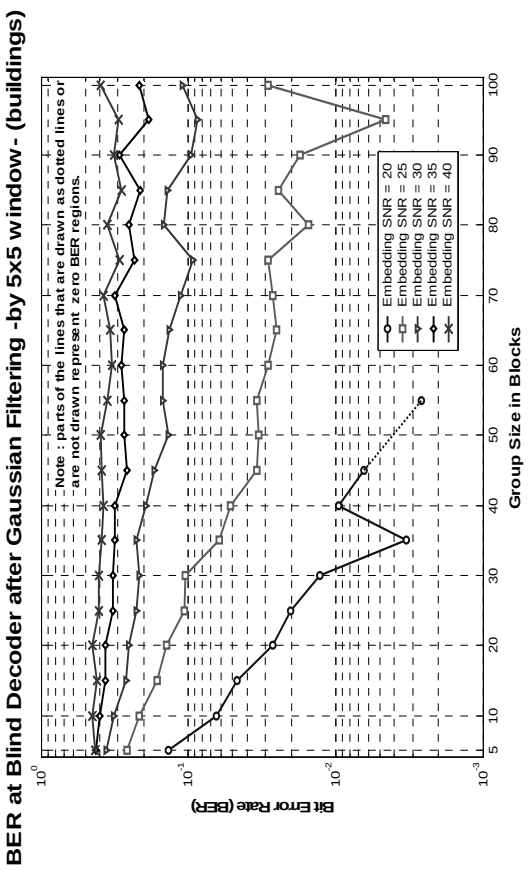


Fig. 3.13 (d)

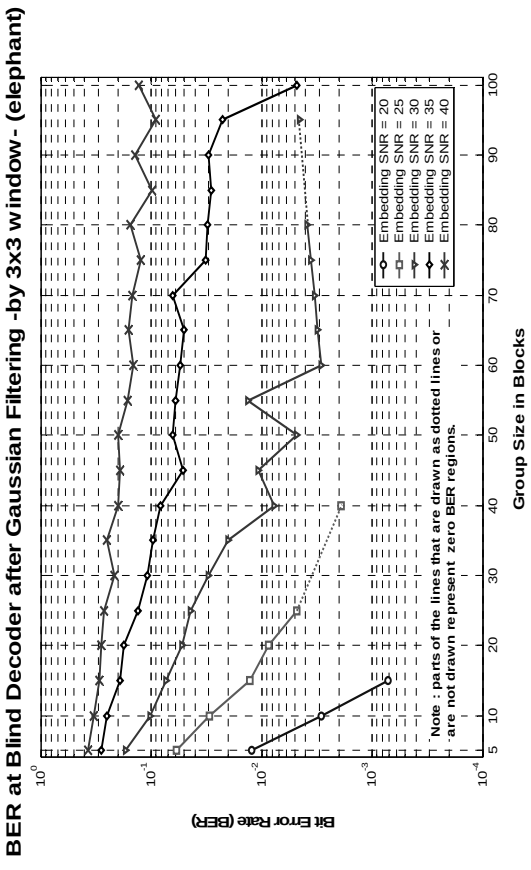


Fig. 3.13 (e)

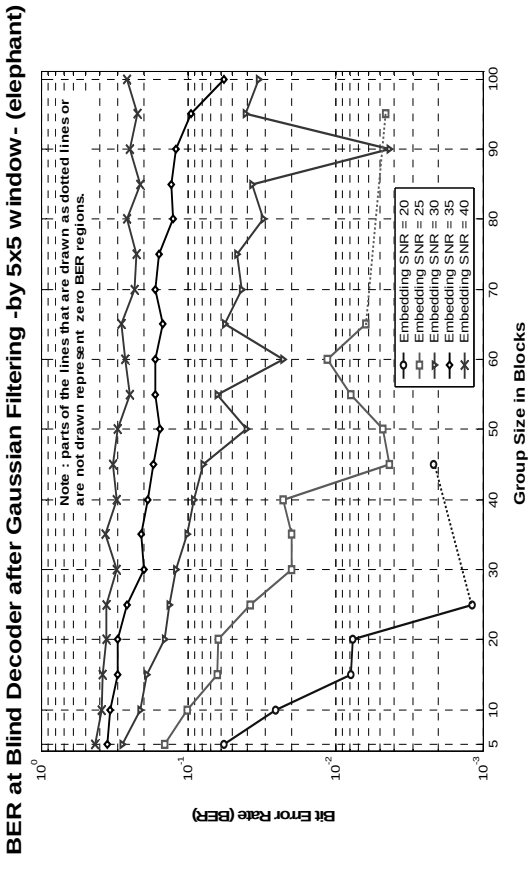


Fig. 3.13 (f)

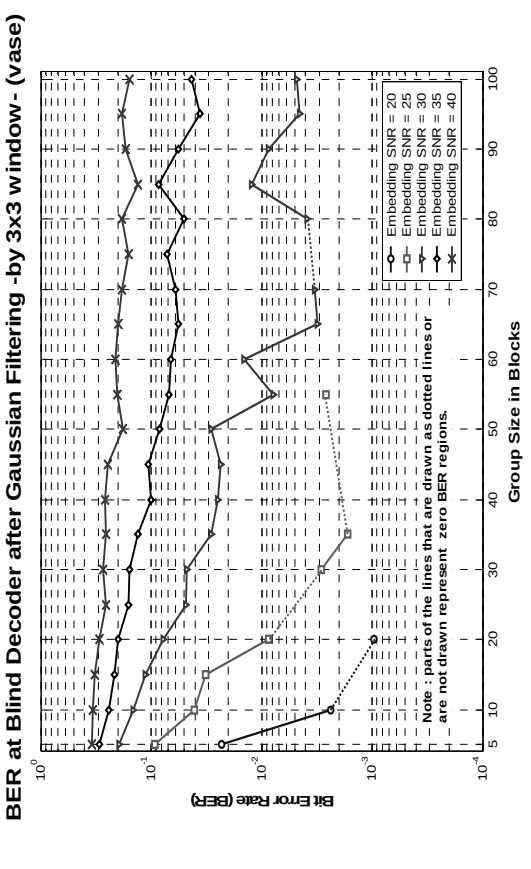


Fig. 3.13 (g)

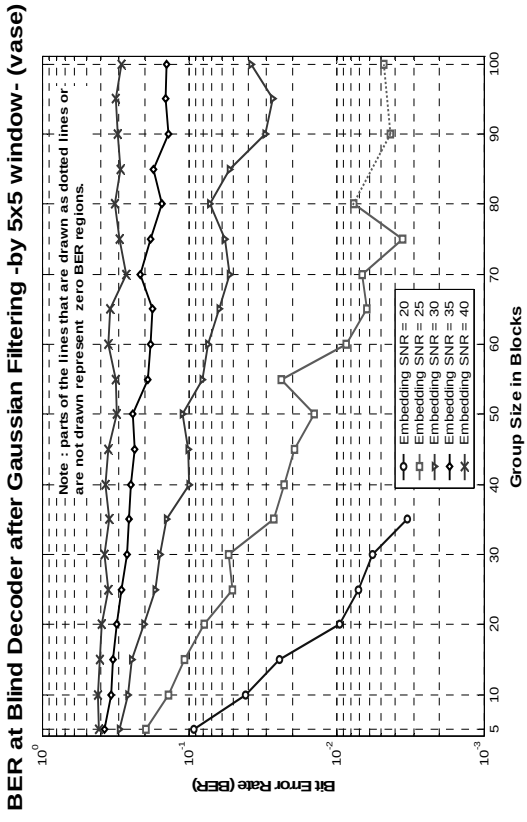


Fig. 3.13 (h)

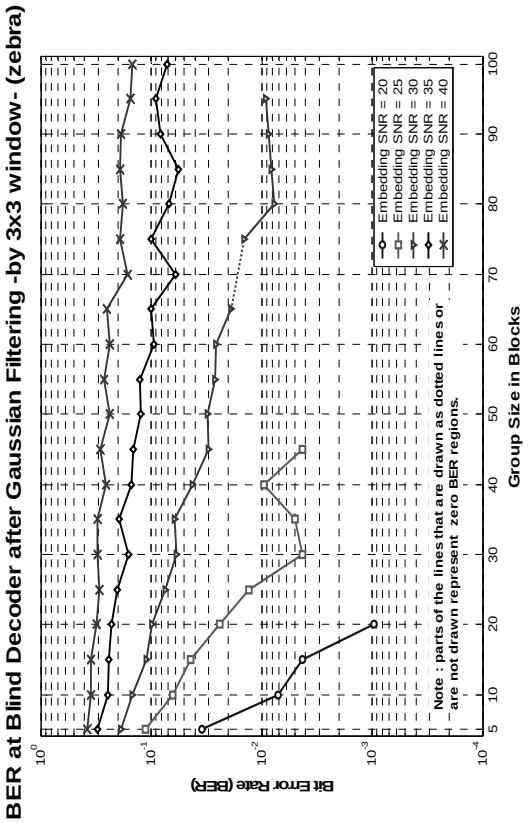


Fig. 3.13 (i)

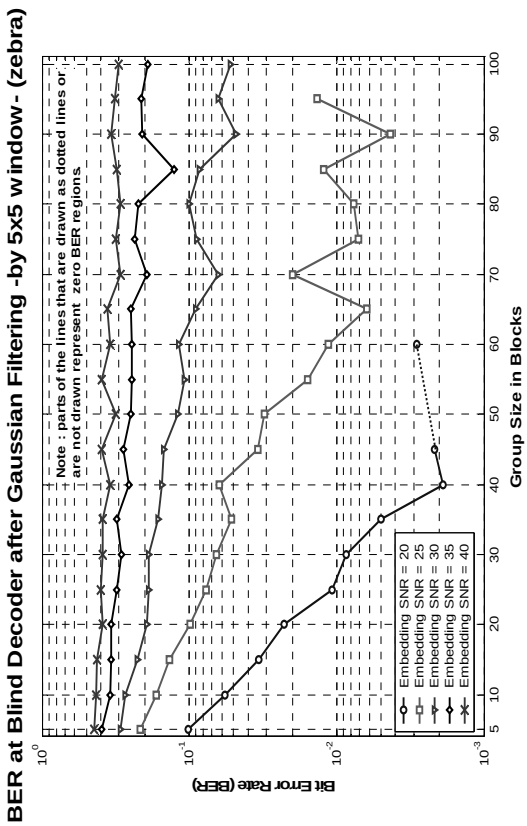


Fig. 3.13 (j)

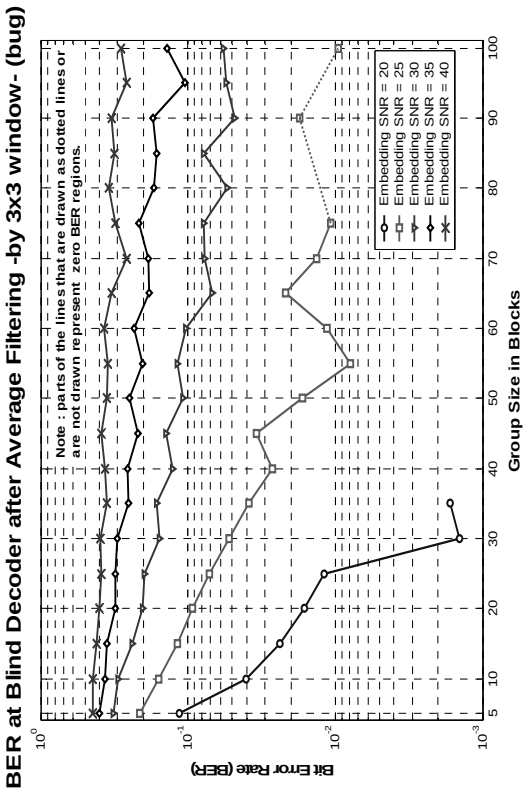


Fig. 3.13 (k)

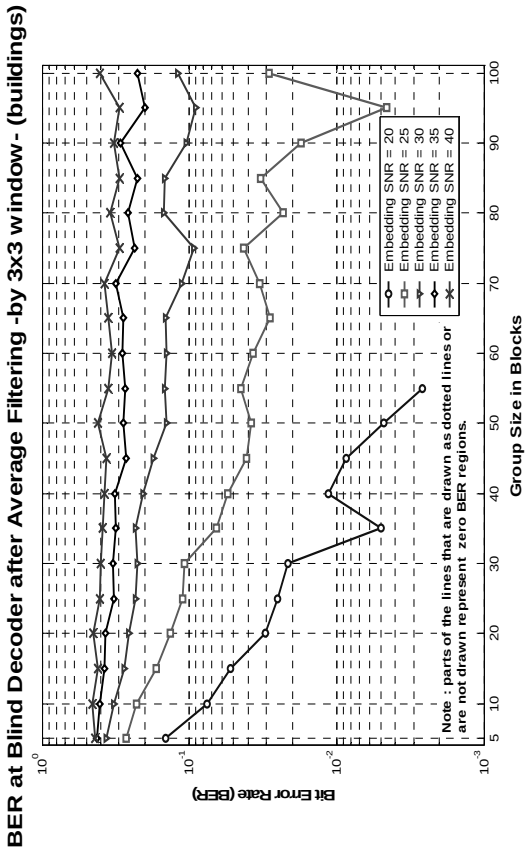


Fig. 3.13 (l)

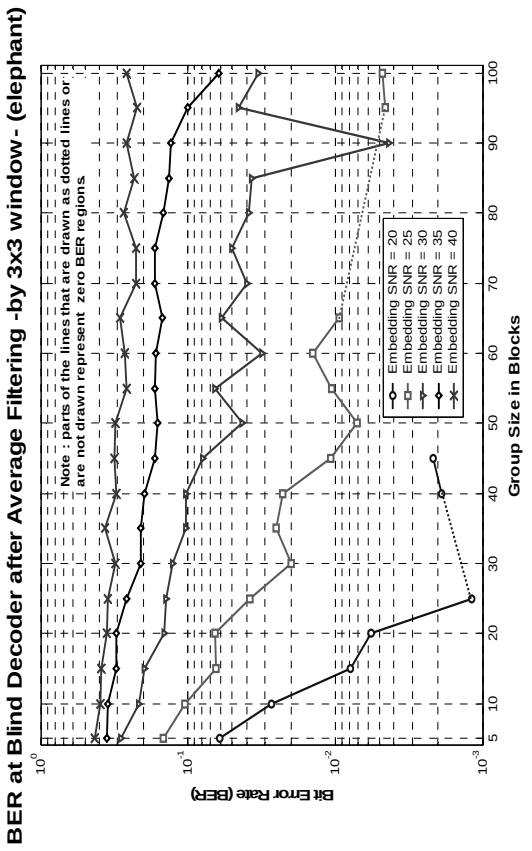


Fig. 3.13 (m)

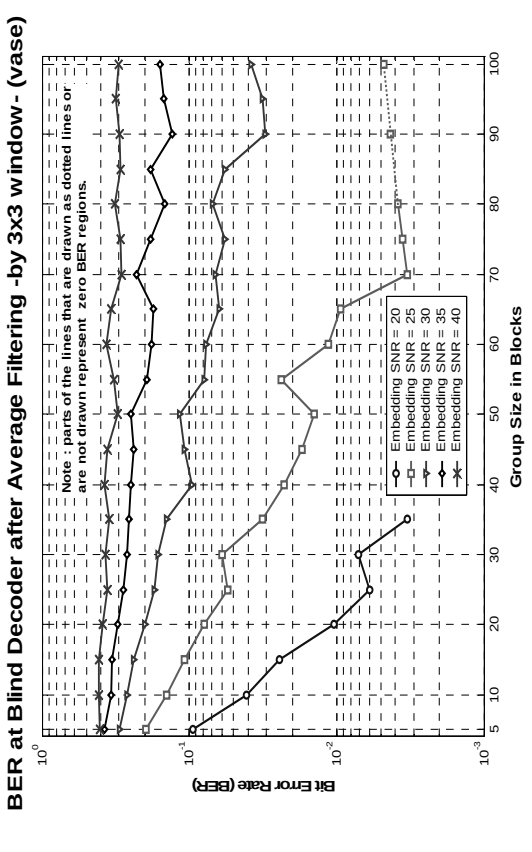


Fig. 3.13 (n)

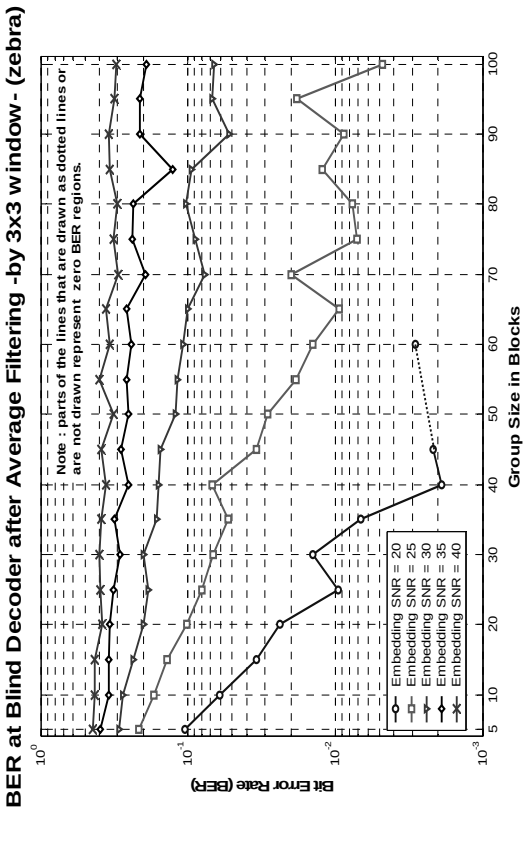


Fig. 3.13 (o)

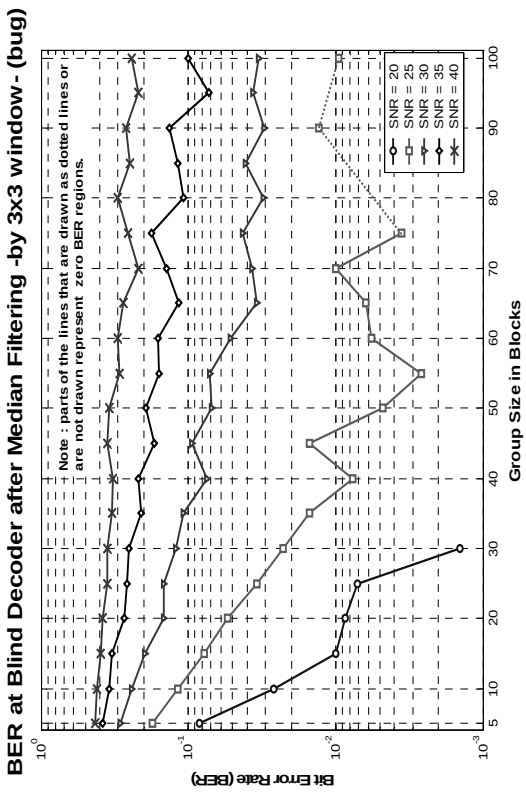


Fig. 3.13 (p)

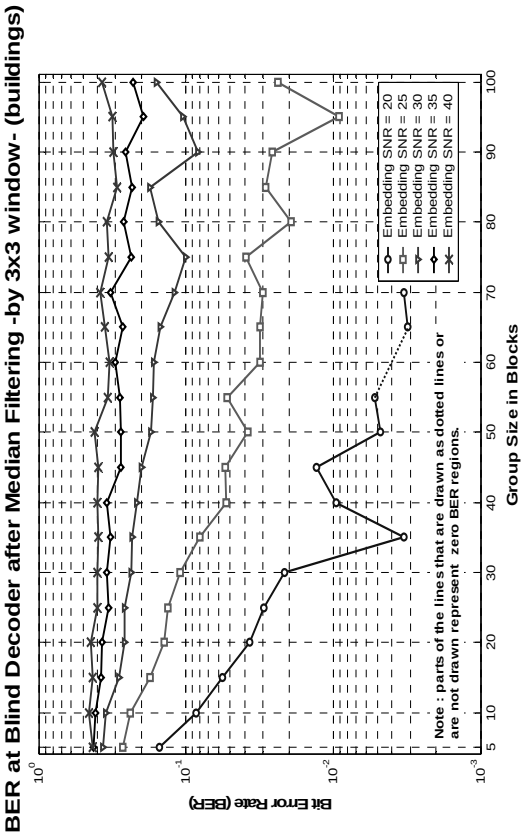


Fig. 3.13 (q)

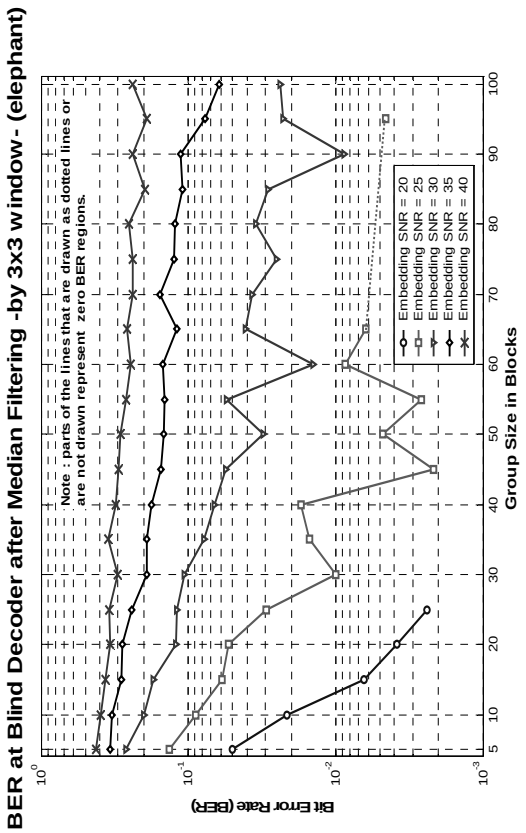


Fig. 3.13 (r)

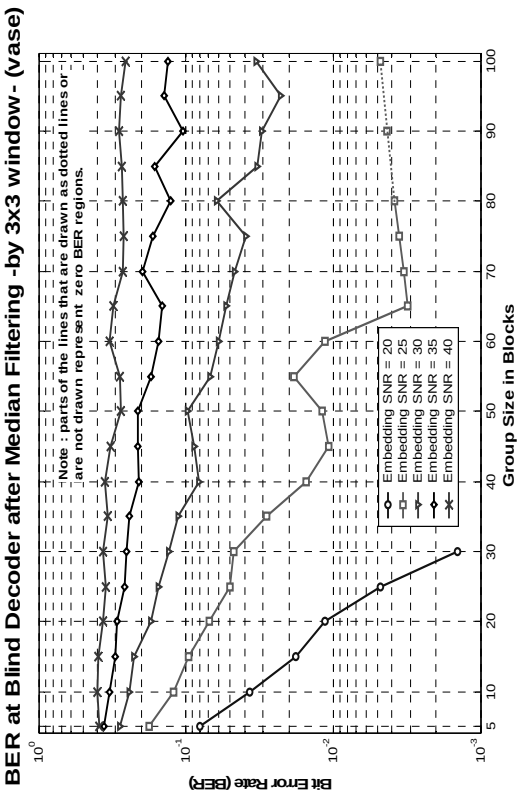


Fig. 3.13 (s)

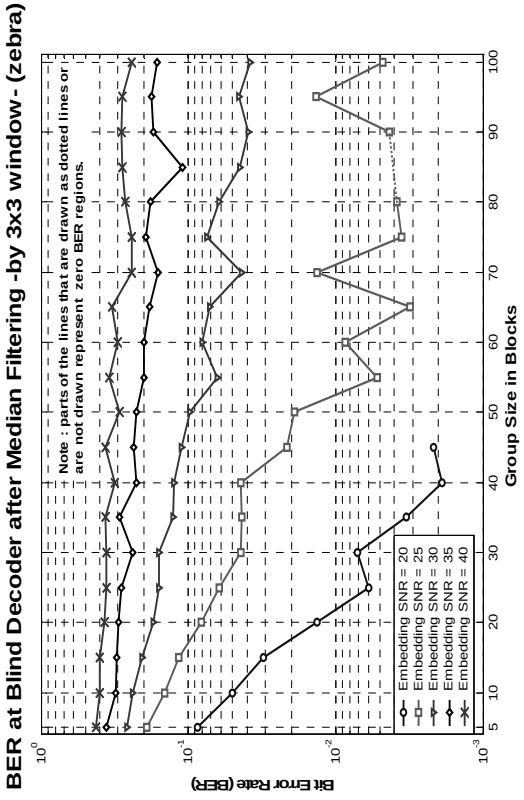


Fig. 3.13 (t)

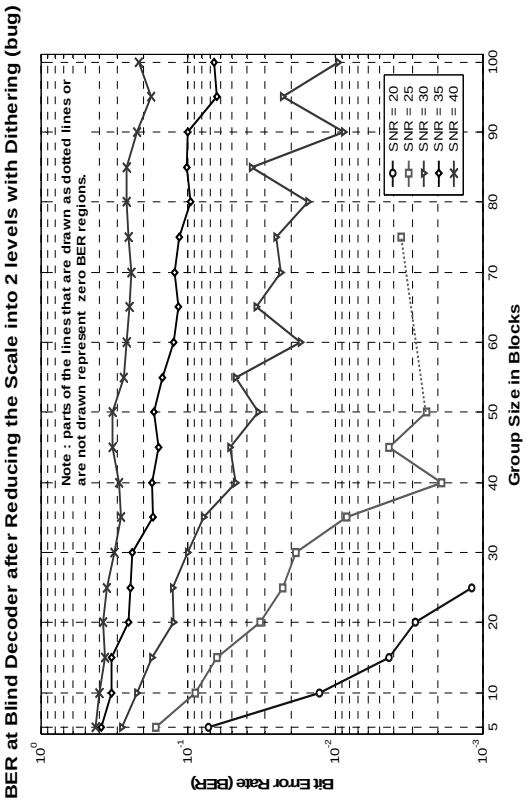


Fig. 3.14 (a)

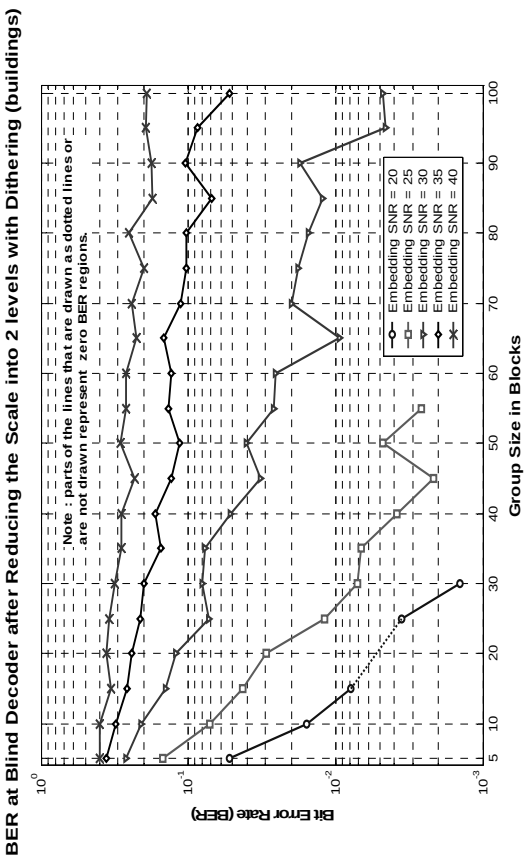


Fig. 3.14 (b)

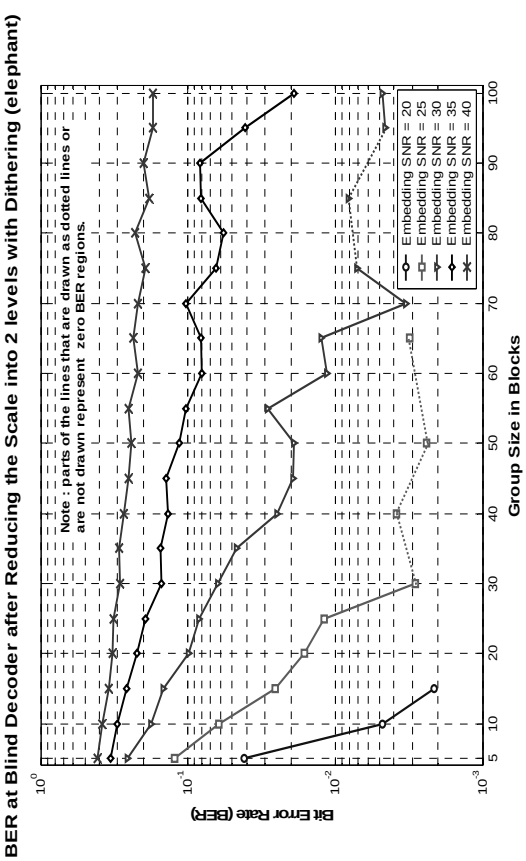


Fig. 3.14 (c)

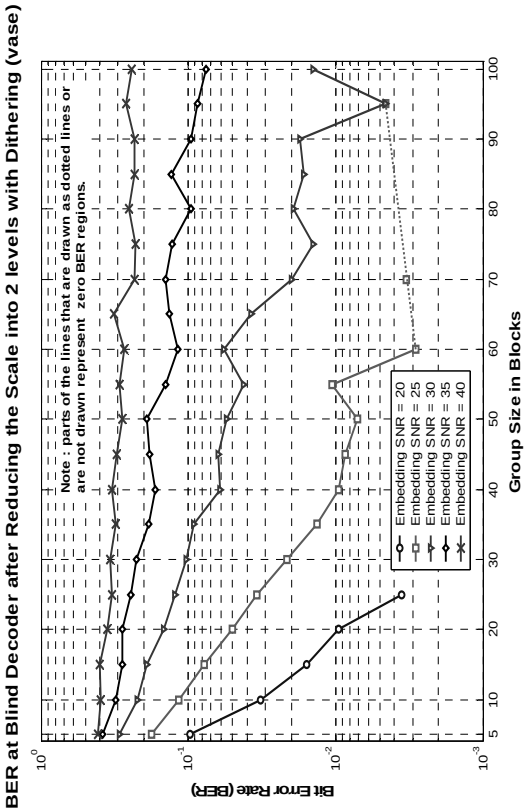


Fig. 3.14 (d)

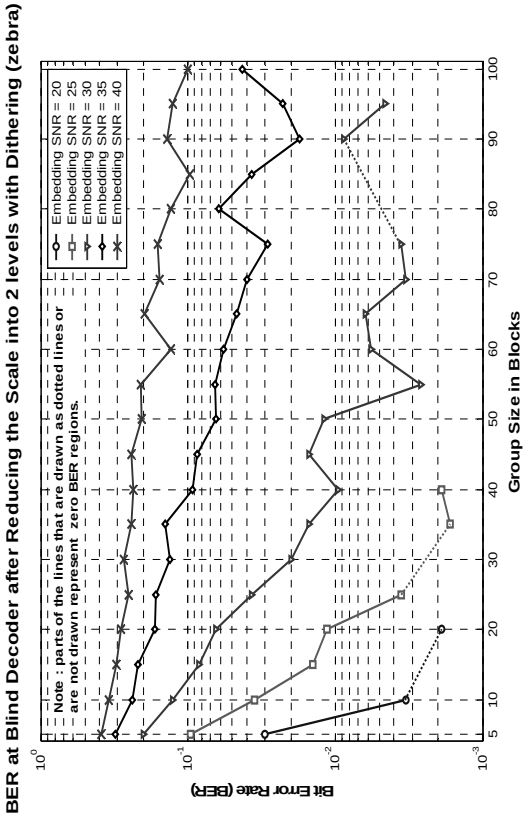


Fig. 3.14 (e)

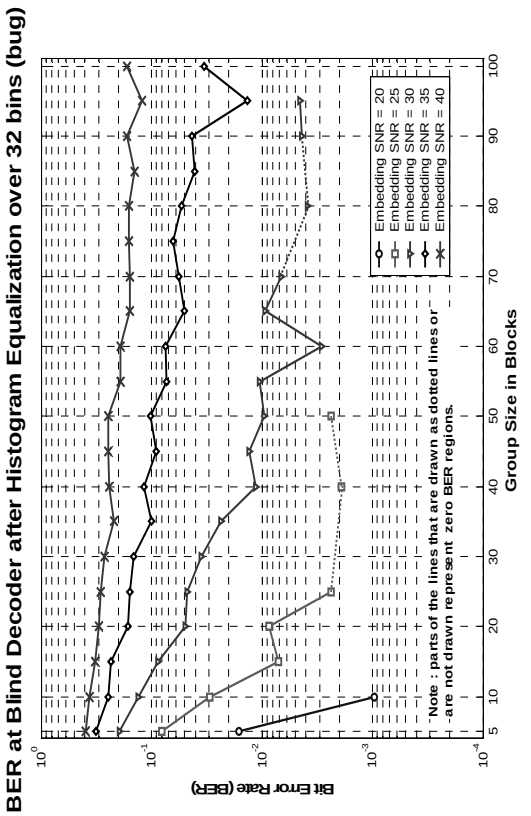


Fig. 3.15 (a)

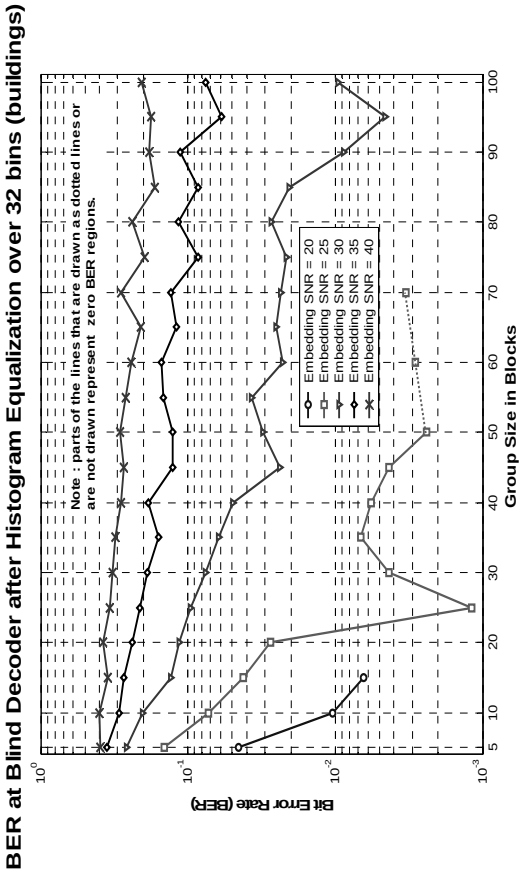


Fig. 3.15 (b)

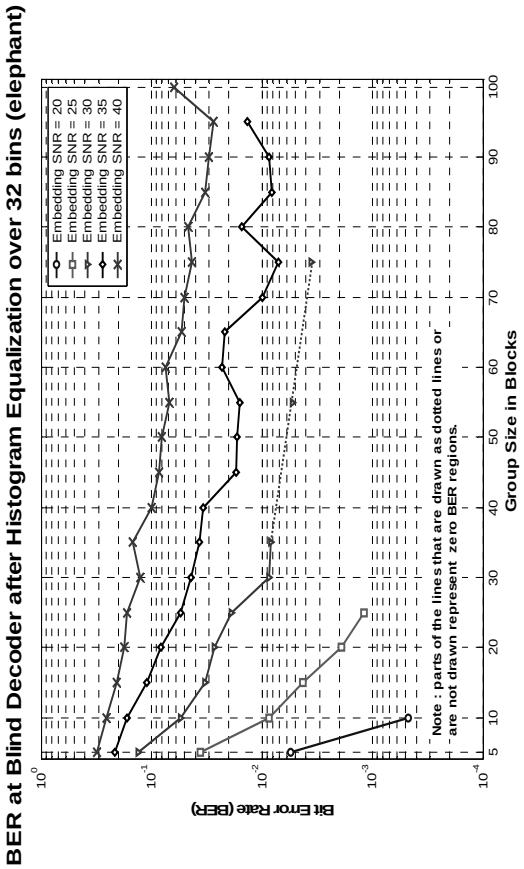


Fig. 3.15 (c)

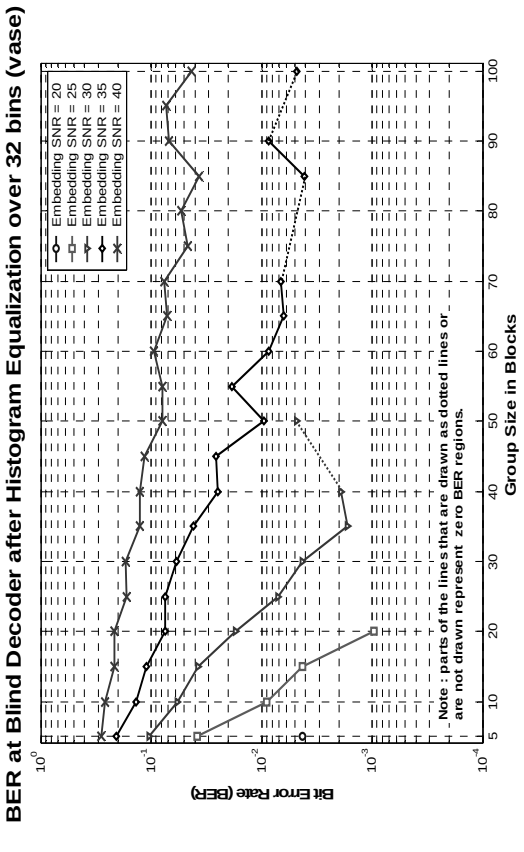


Fig. 3.15 (d)

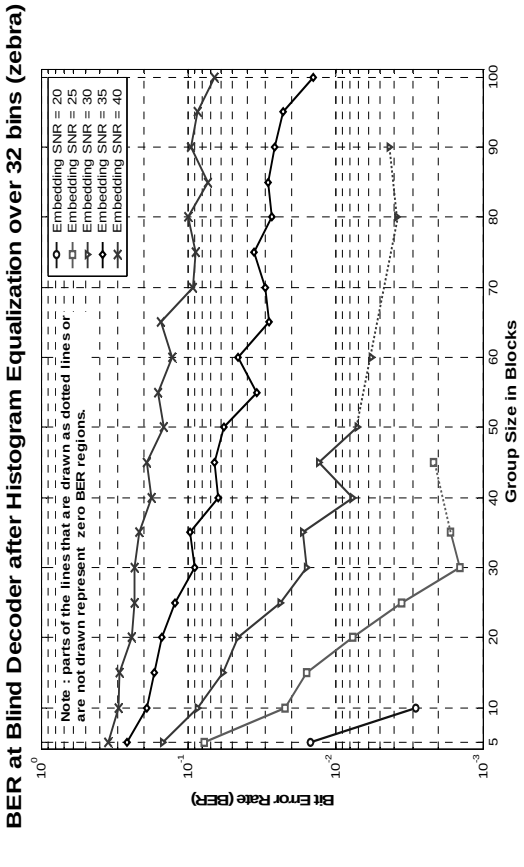


Fig. 3.15 (e)

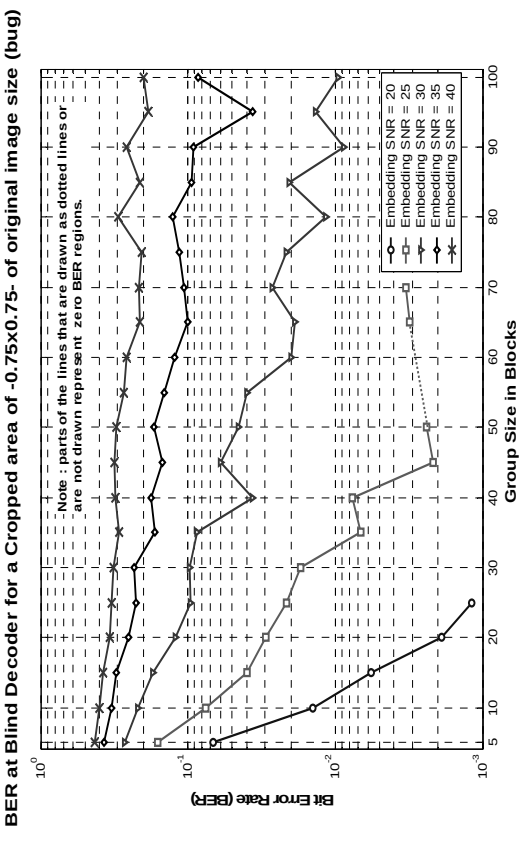


Fig. 3.16 (a)

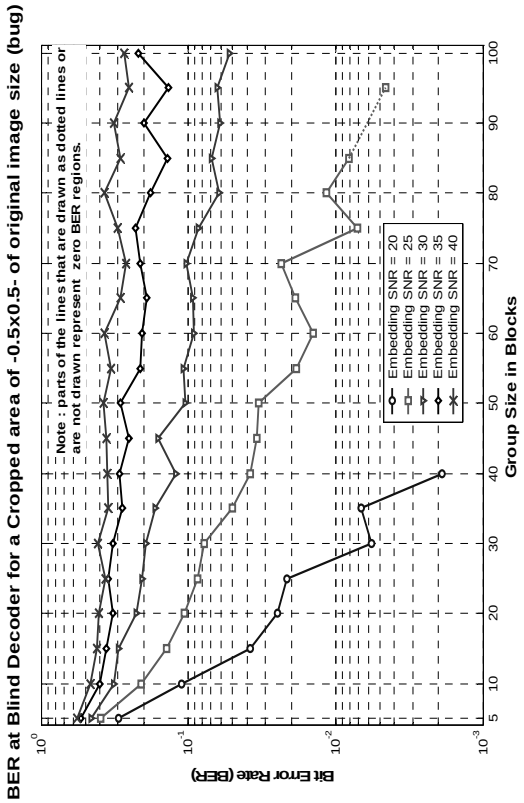


Fig. 3.16 (b)

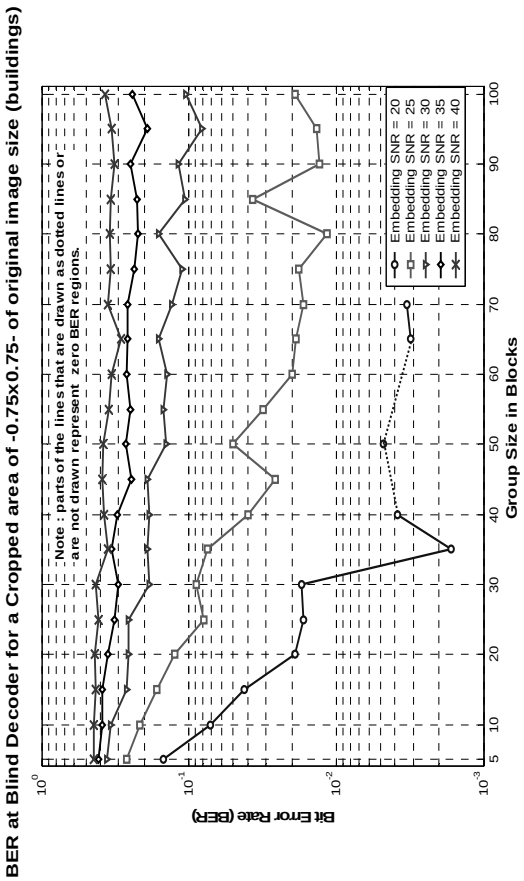


Fig. 3.16 (c)

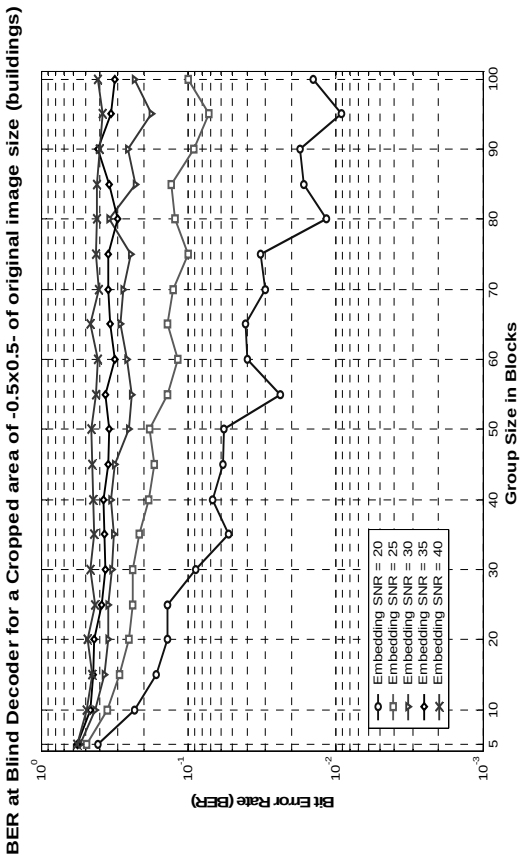


Fig. 3.16 (d)

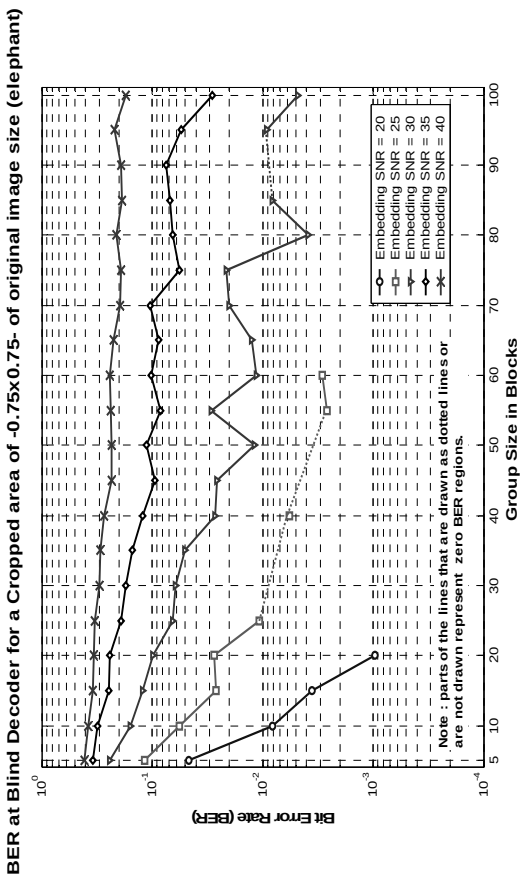


Fig. 3.16 (e)

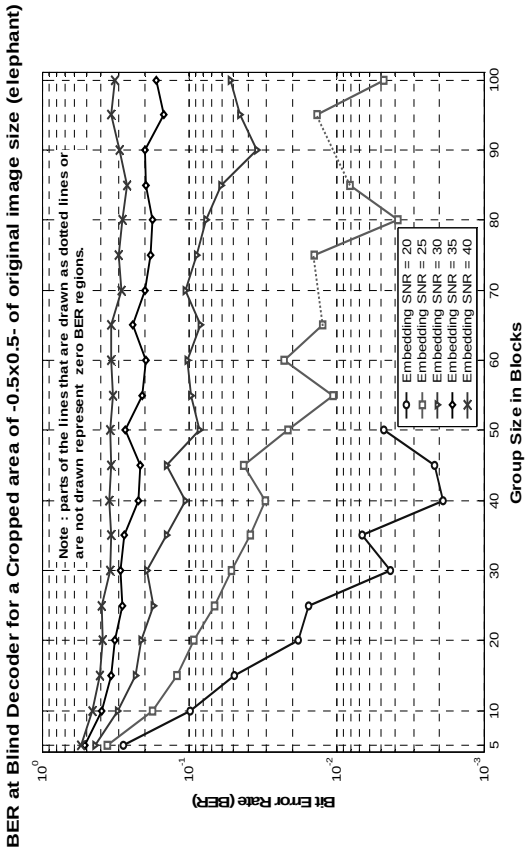


Fig. 3.16 (f)

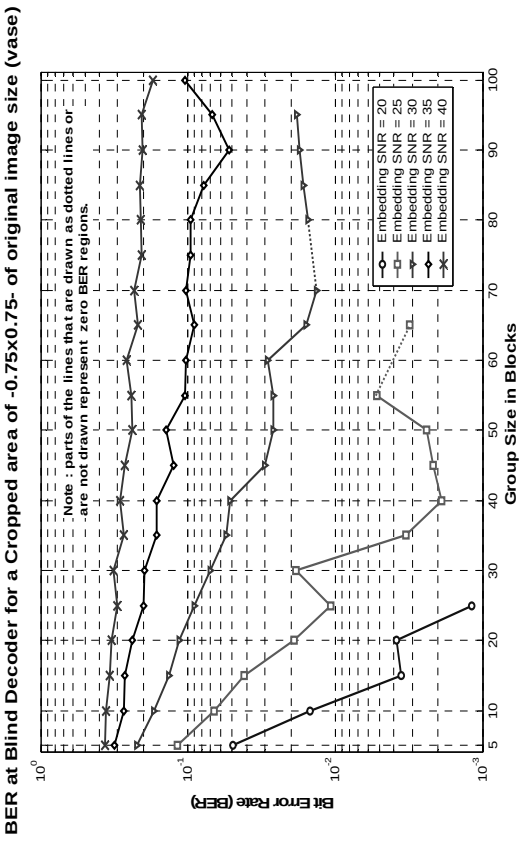


Fig. 3.16 (g)

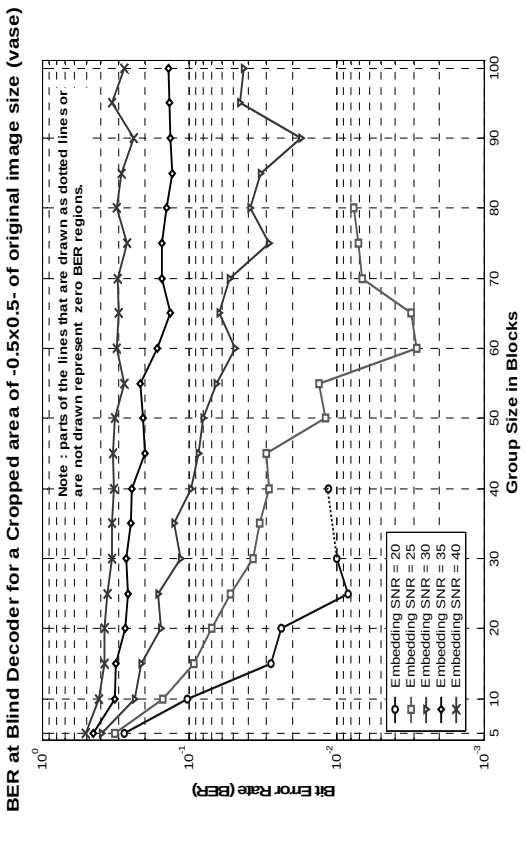


Fig. 3.16 (h)

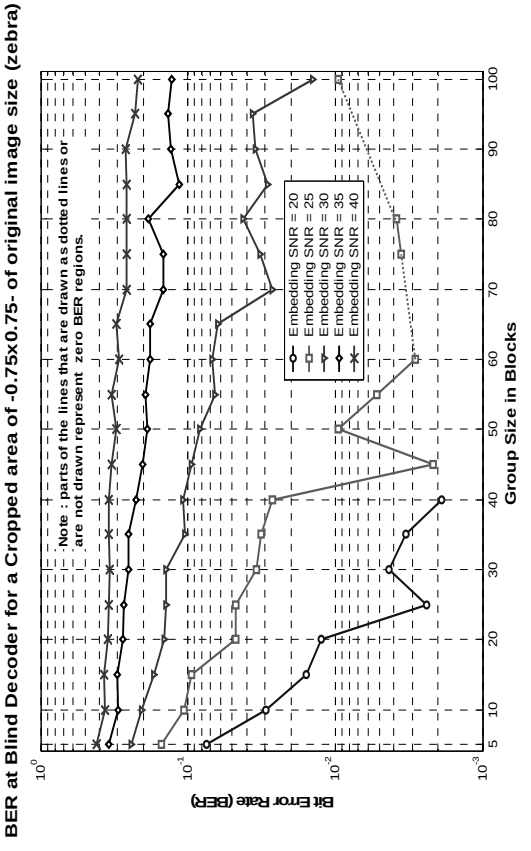


Fig. 3.16 (i)

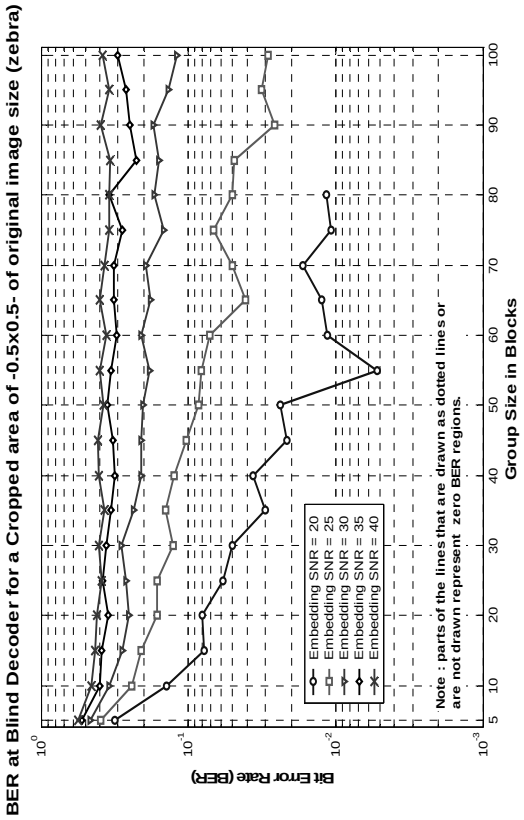


Fig. 3.16 (j)

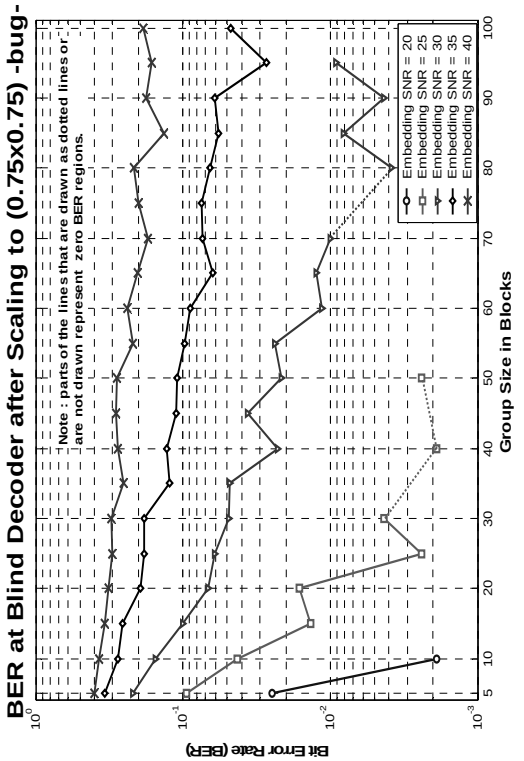


Fig. 3.17 (a)

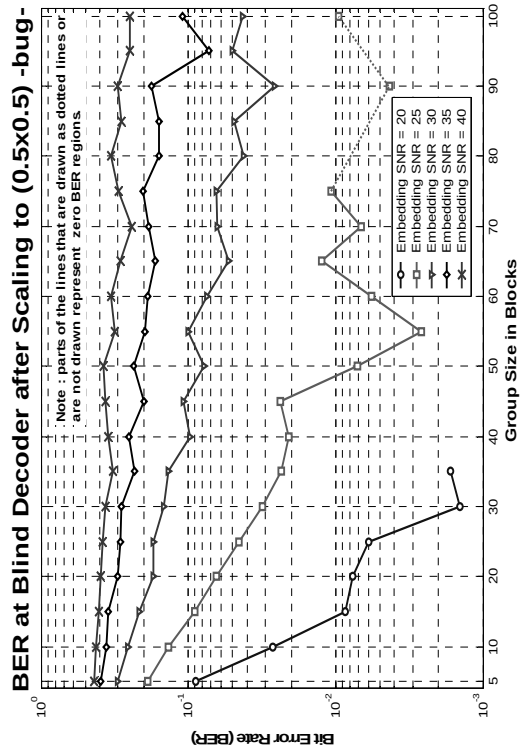


Fig. 3.17 (b)

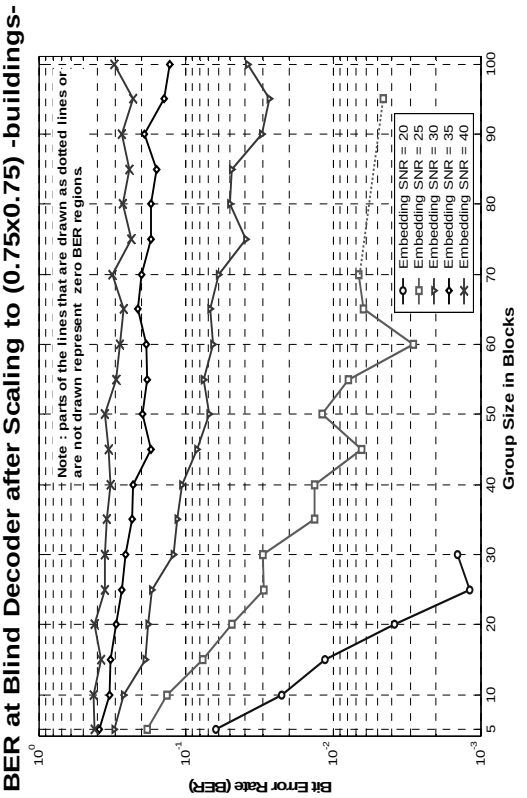


Fig. 3.17 (c)

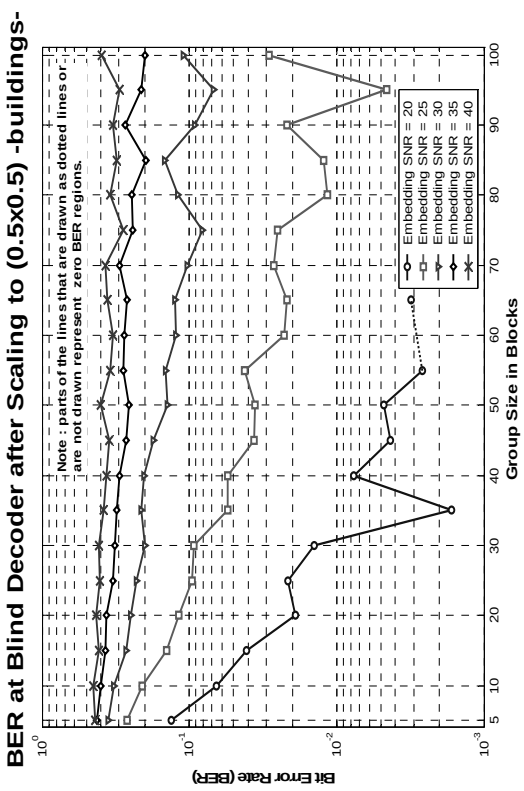


Fig. 3.17 (d)

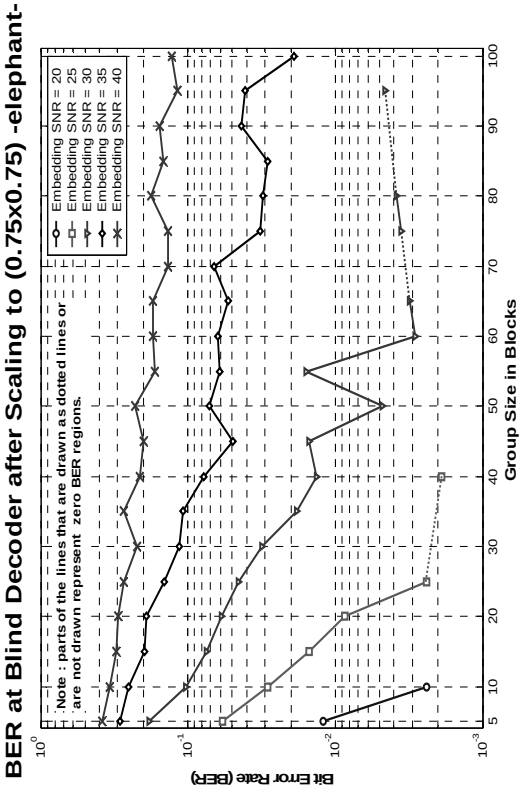


Fig. 3.17 (e)

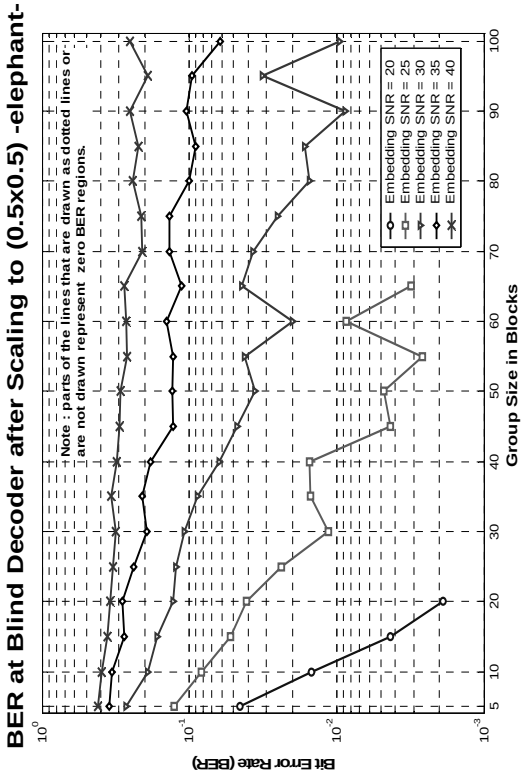


Fig. 3.17 (f)

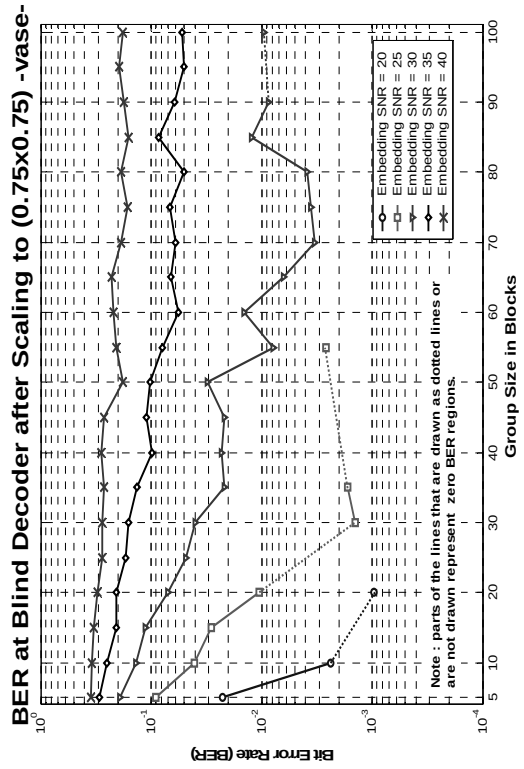


Fig. 3.17 (g)

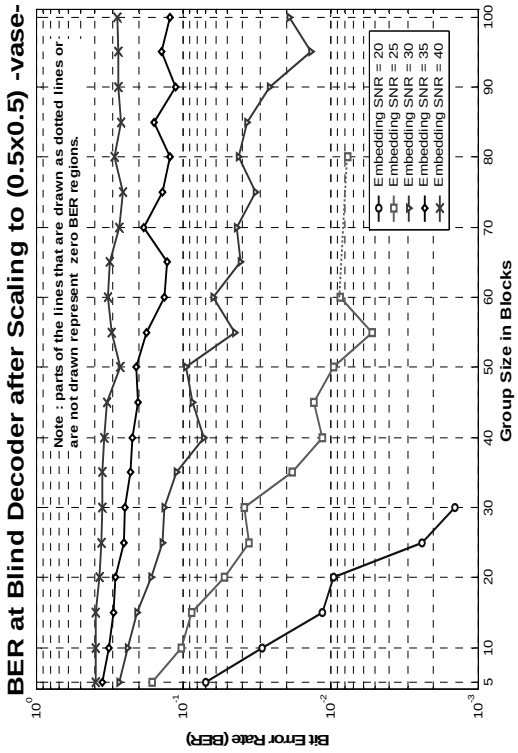


Fig. 3.17 (h)

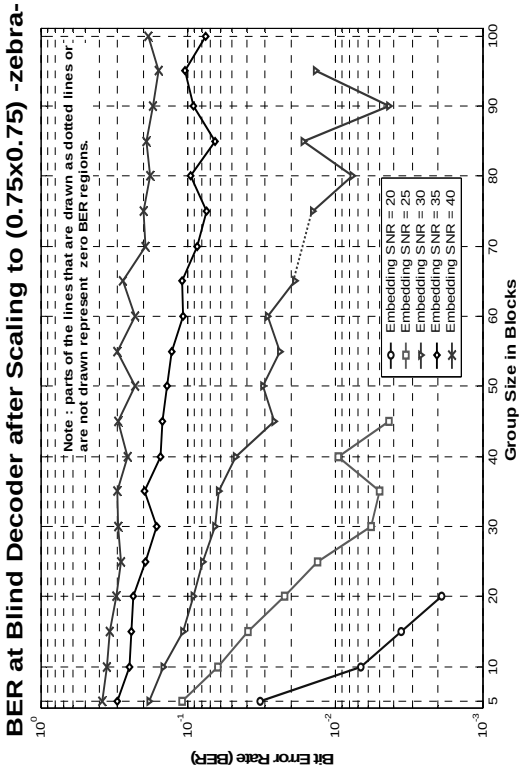


Fig. 3.17 (i)

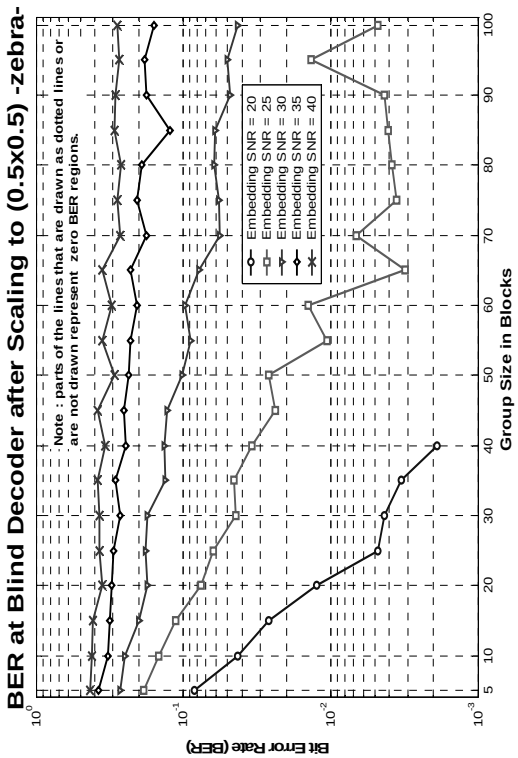


Fig. 3.17 (j)

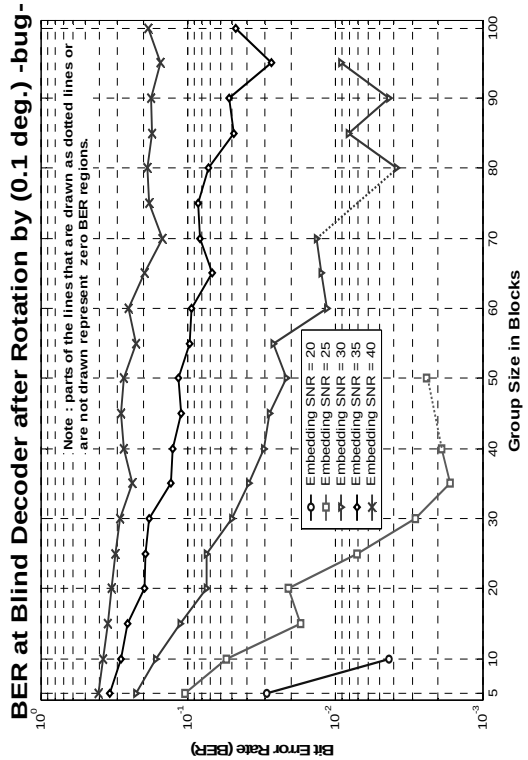


Fig. 3.18 (a)

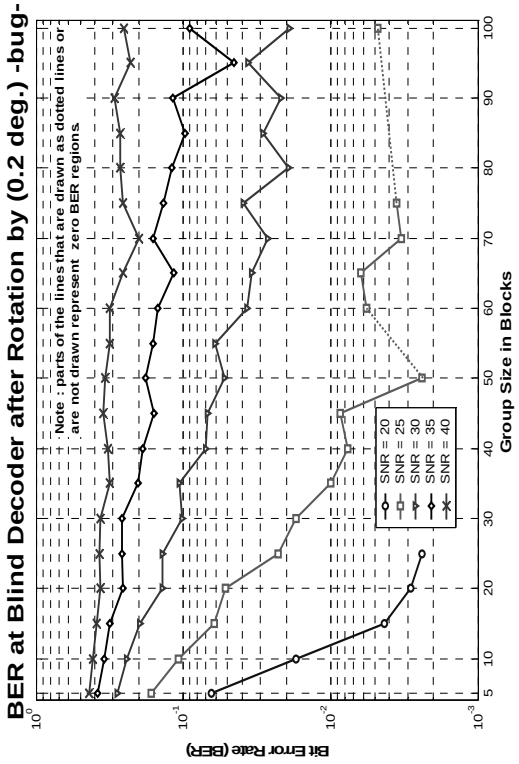


Fig. 3.18 (b)

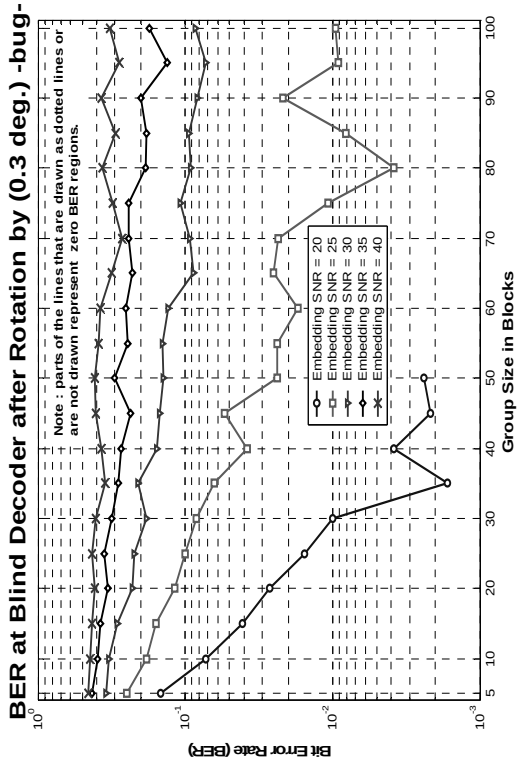


Fig. 3.18 (c)

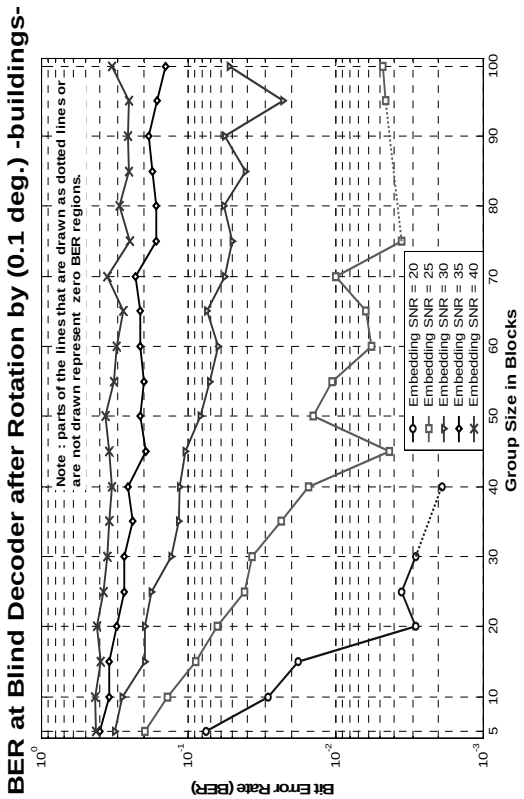


Fig. 3.18 (d)

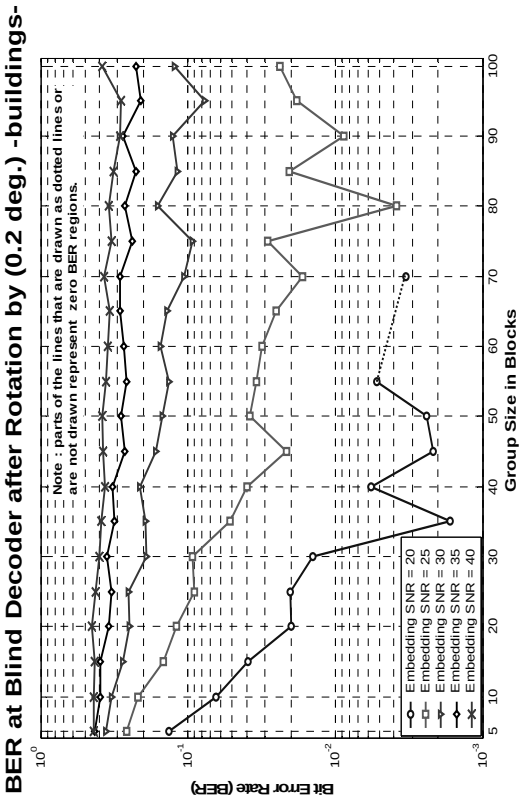


Fig. 3.18 (e)

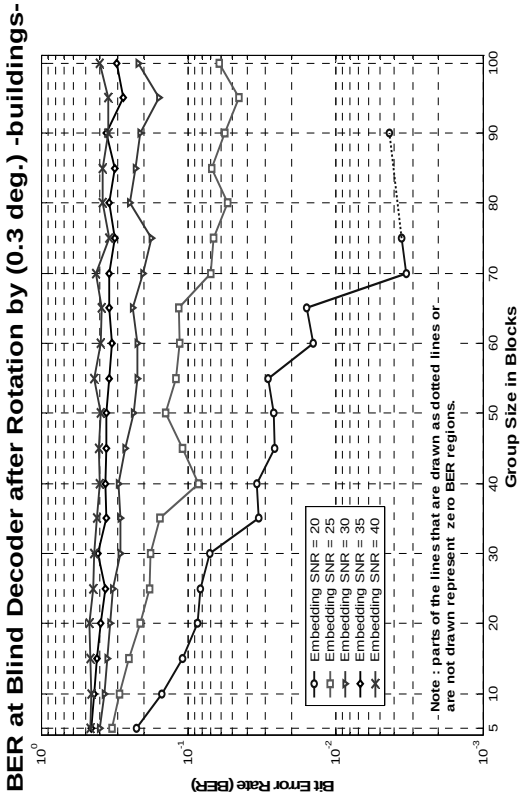


Fig. 3.18 (f)

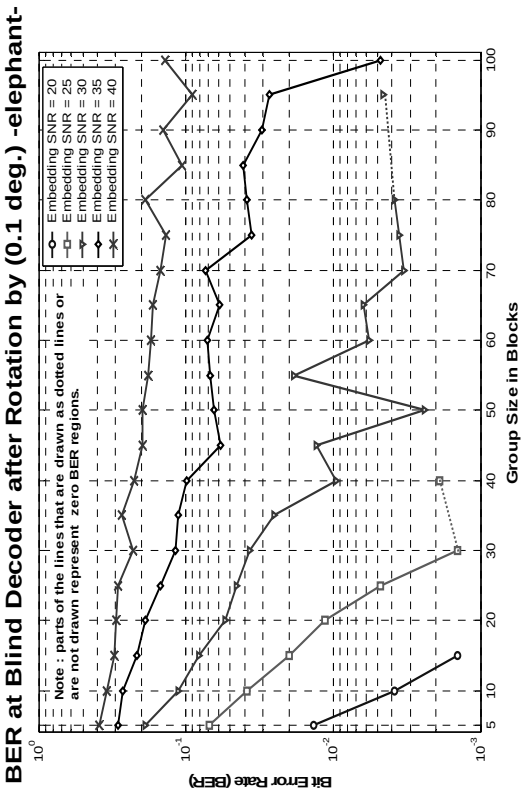


Fig. 3.18 (g)

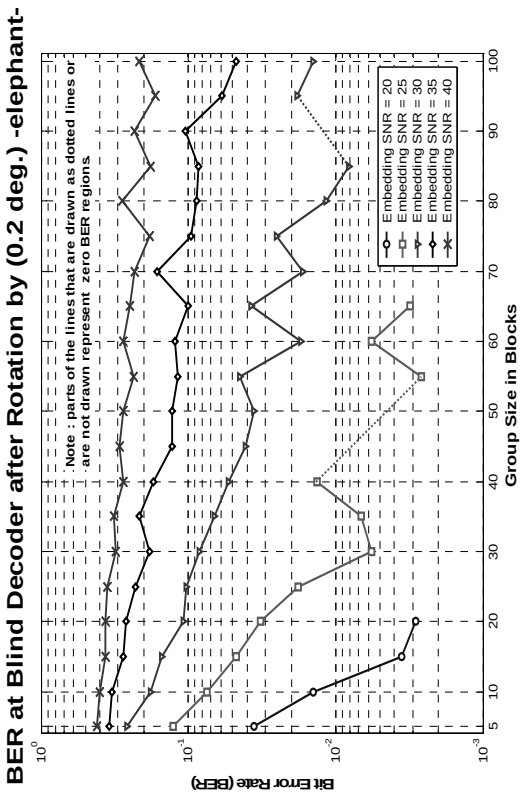


Fig. 3.18 (h)

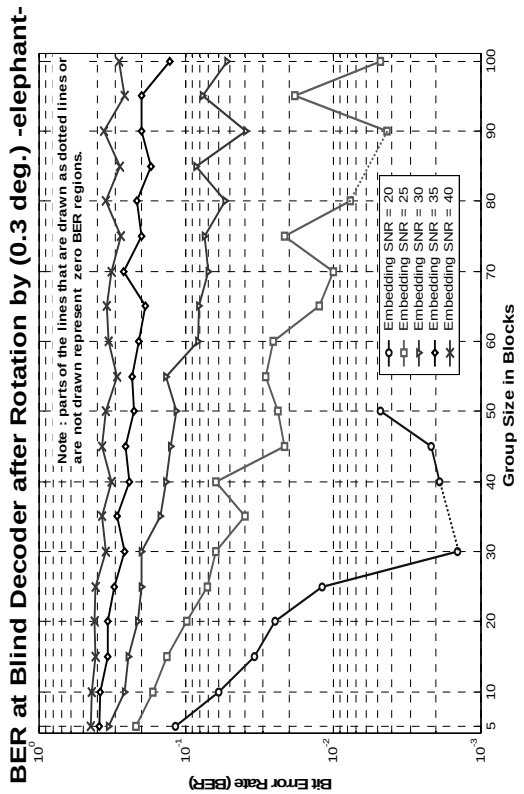


Fig. 3.18 (i)

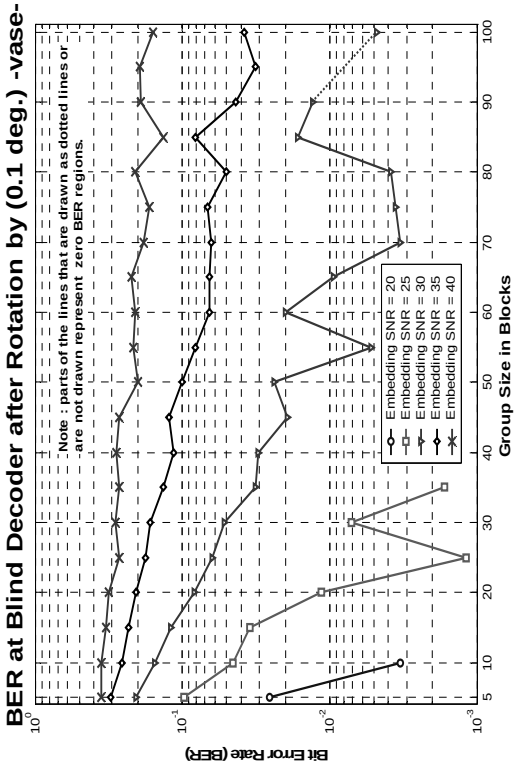


Fig. 3.18 (j)

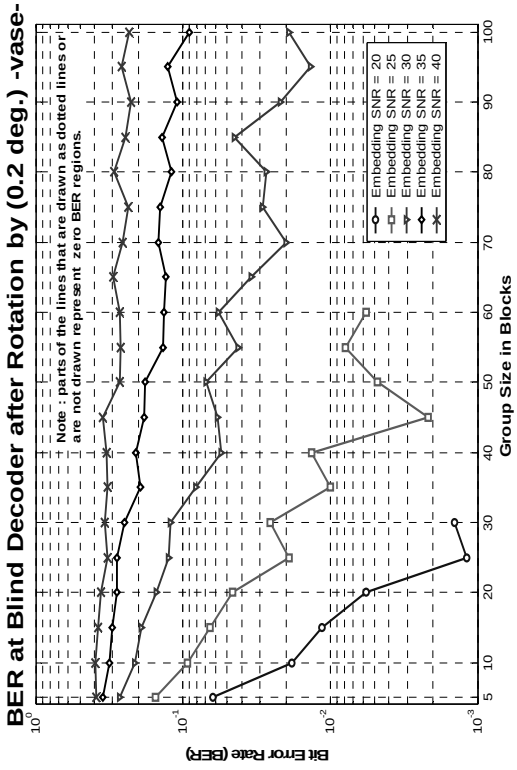


Fig. 3.18 (k)

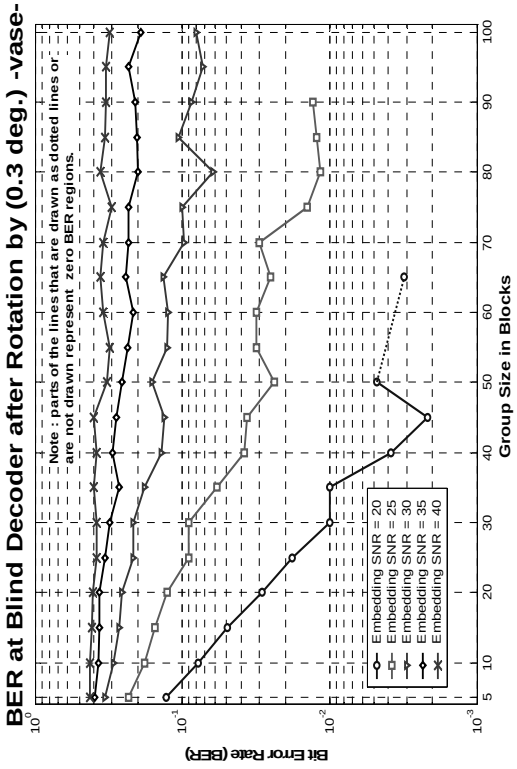


Fig. 3.18 (l)

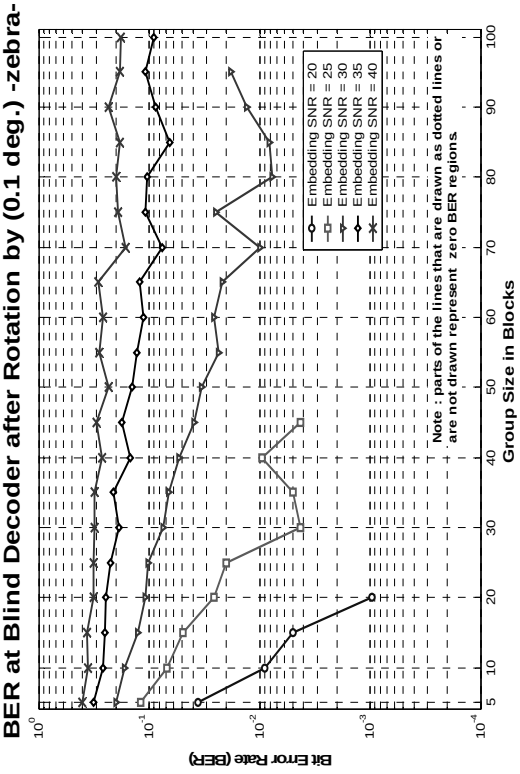


Fig. 3.18 (m)

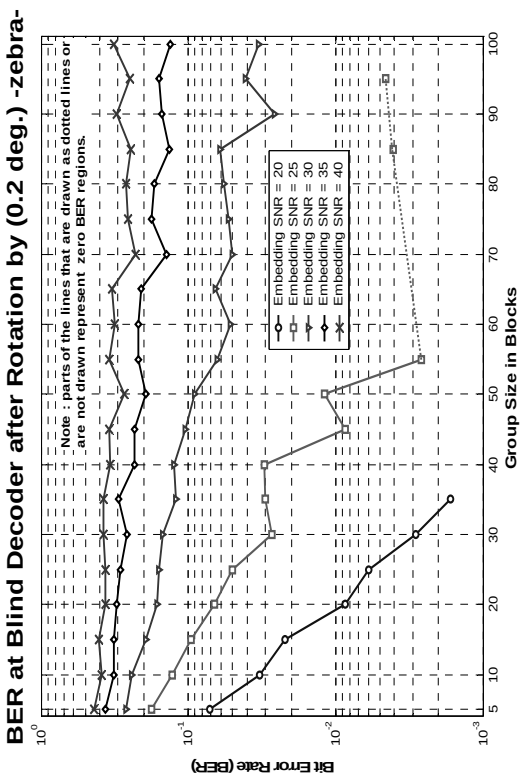


Fig. 3.18 (n)

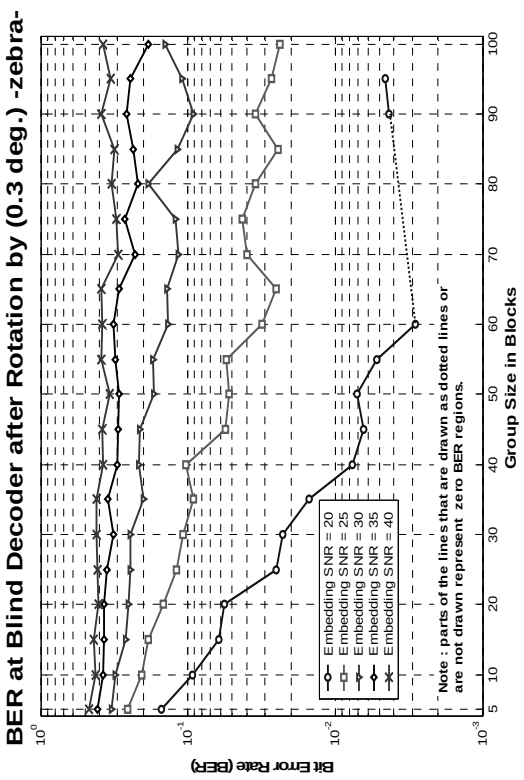


Fig. 3.18 (o)

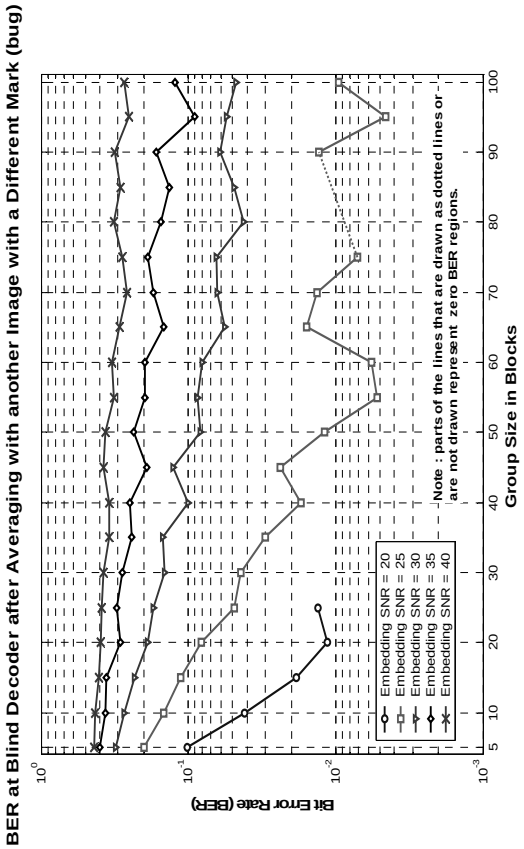


Fig. 3.19 (a)

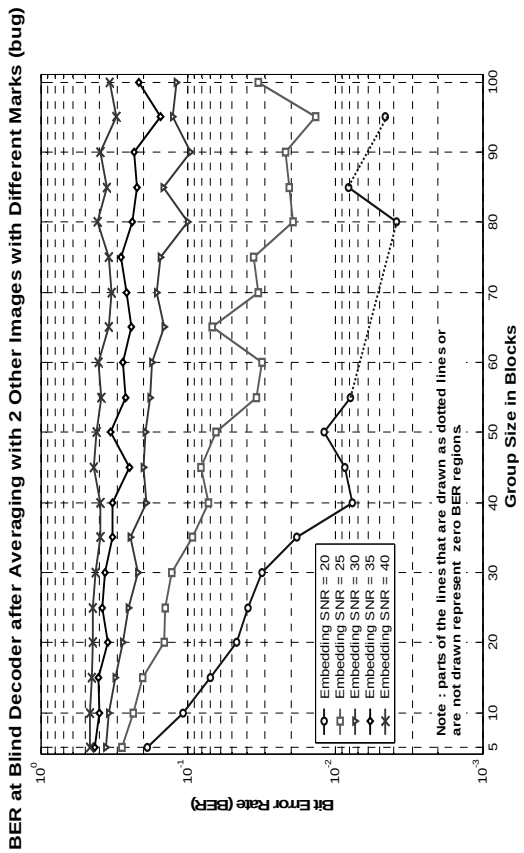


Fig. 3.19 (b)

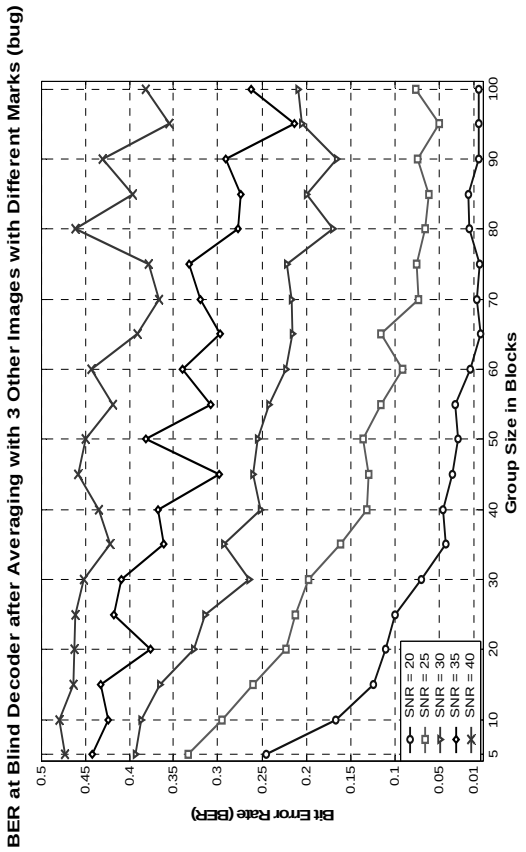


Fig. 3.19 (c)

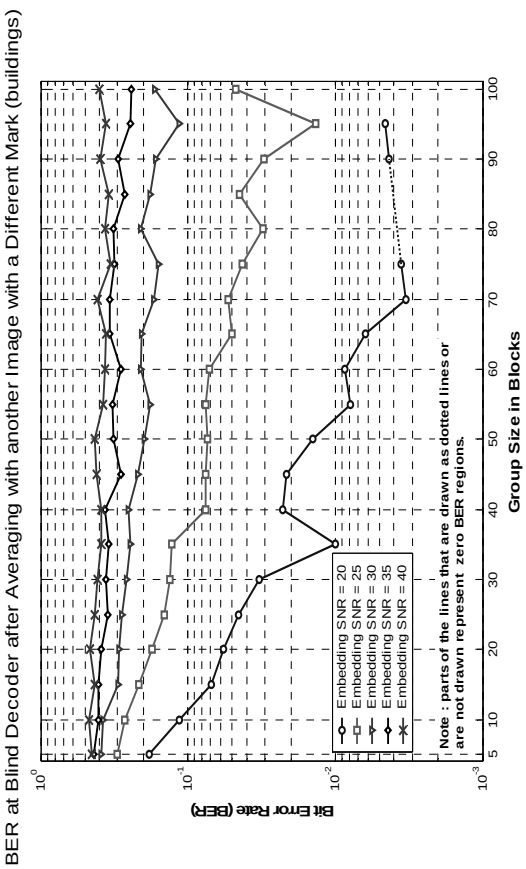


Fig. 3.19 (d)

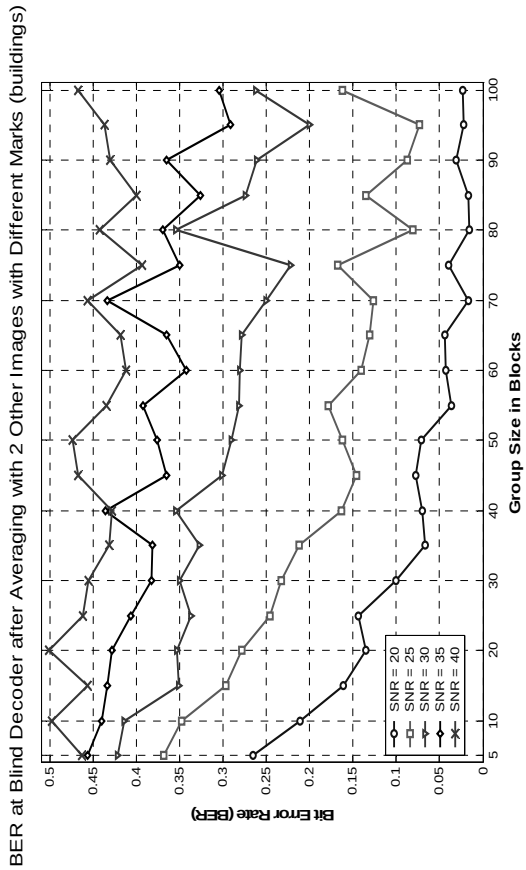


Fig. 3.19 (e)

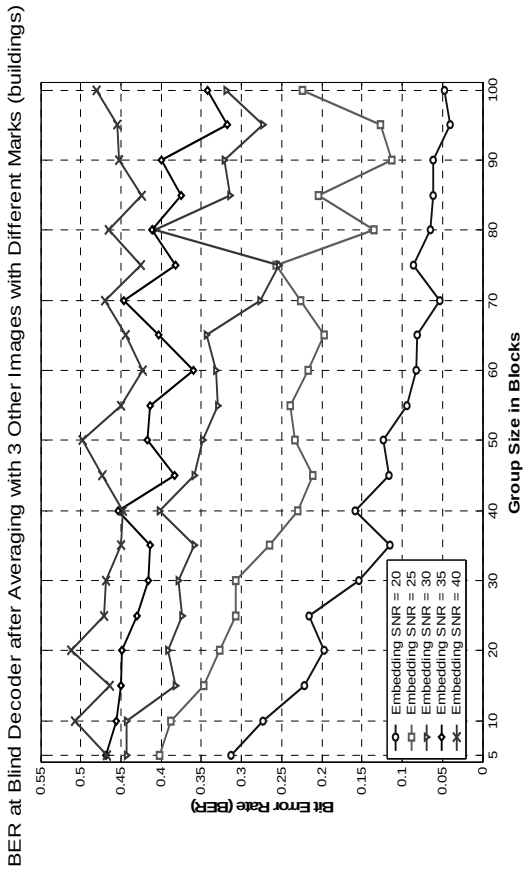


Fig. 3.19 (f)

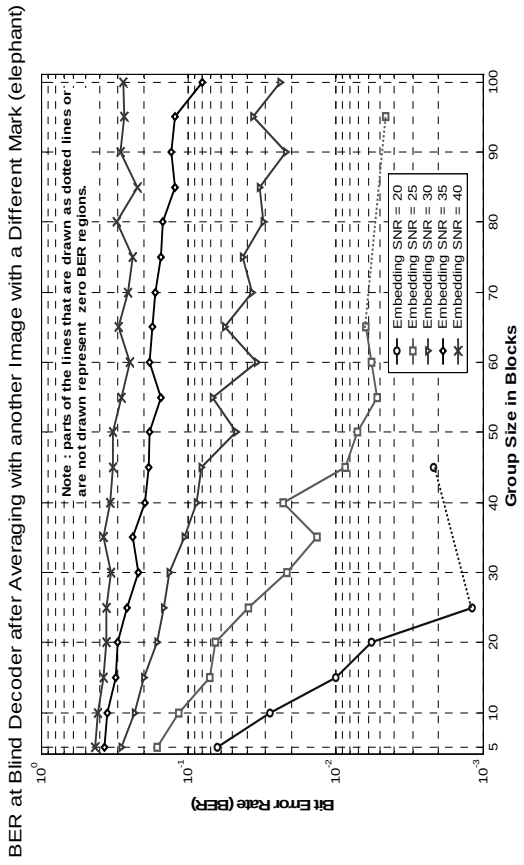


Fig. 3.19 (g)

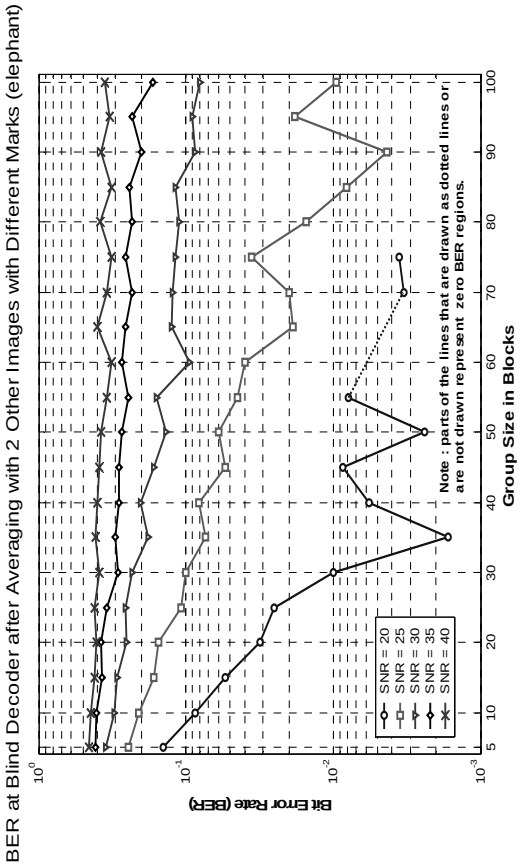


Fig. 3.19 (h)

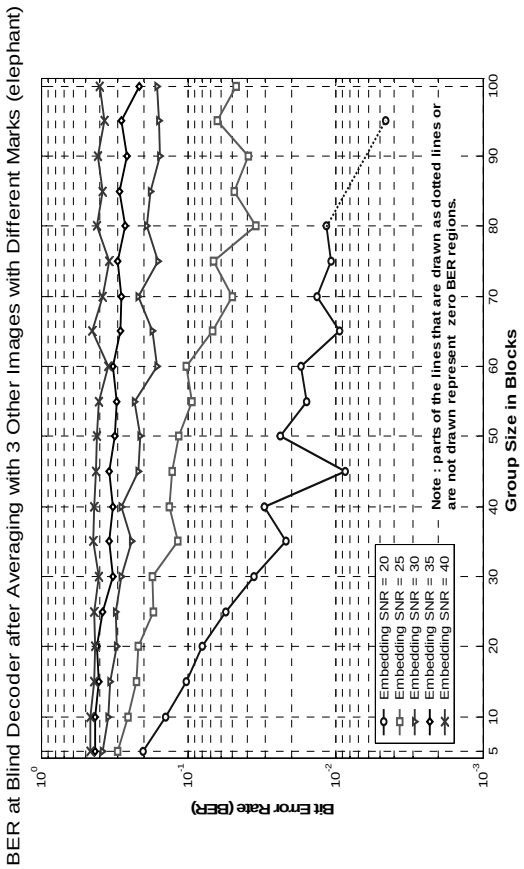


Fig. 3.19 (i)

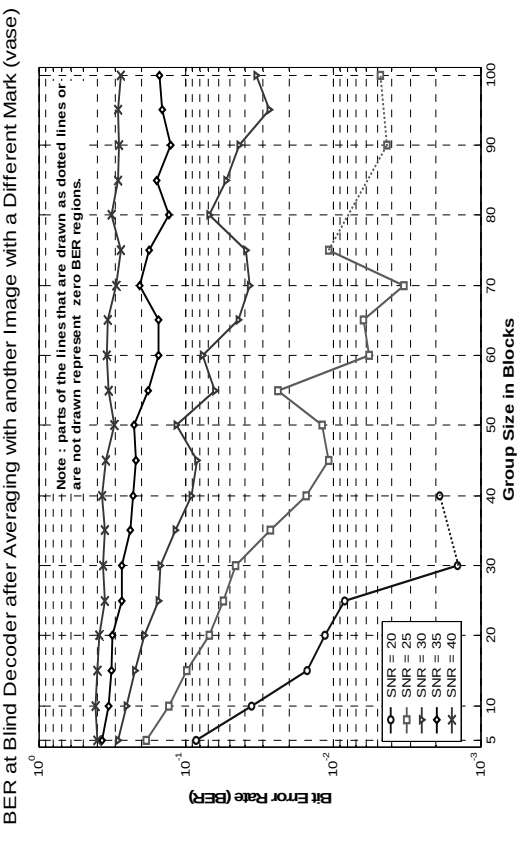


Fig. 3.19 (j)

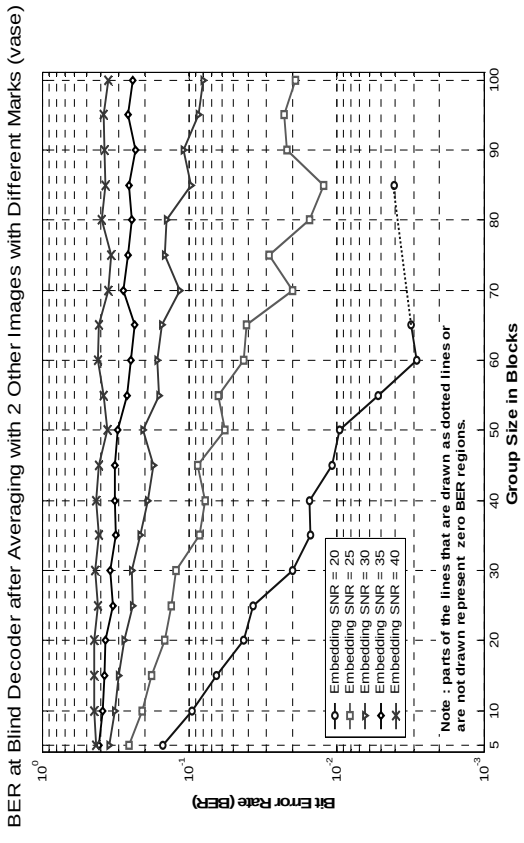


Fig. 3.19 (k)

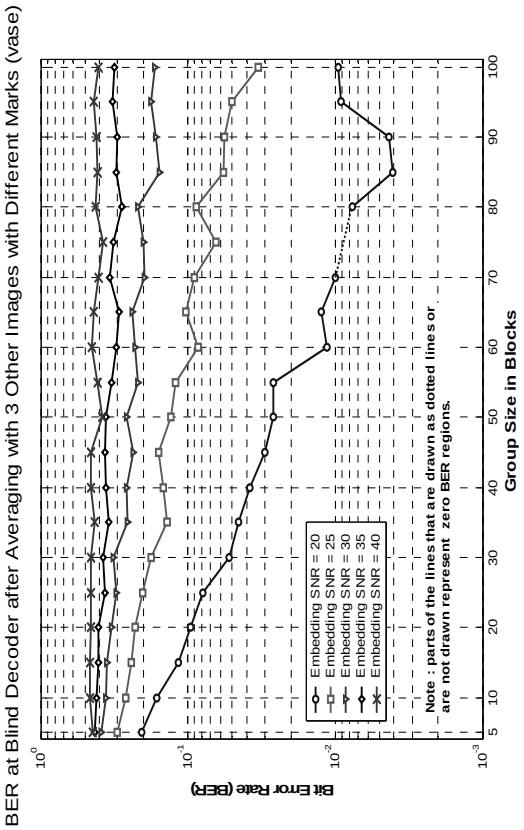


Fig. 3.19 (l)

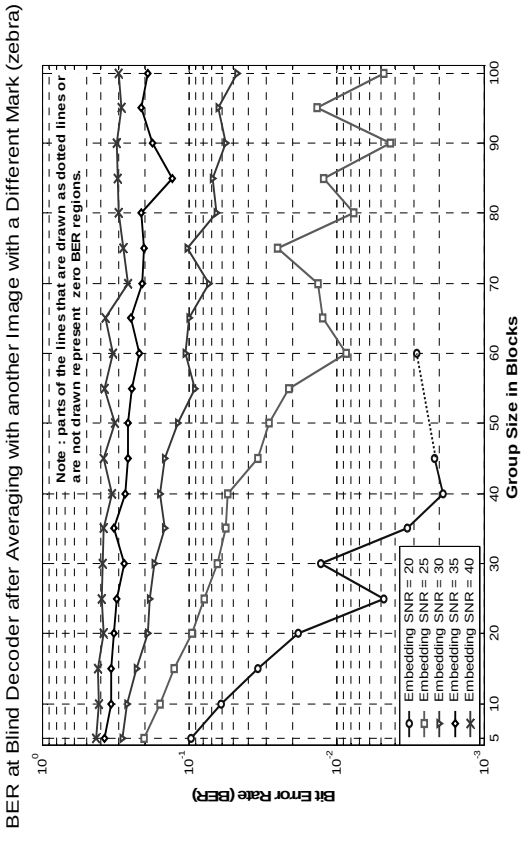


Fig. 3.19 (m)

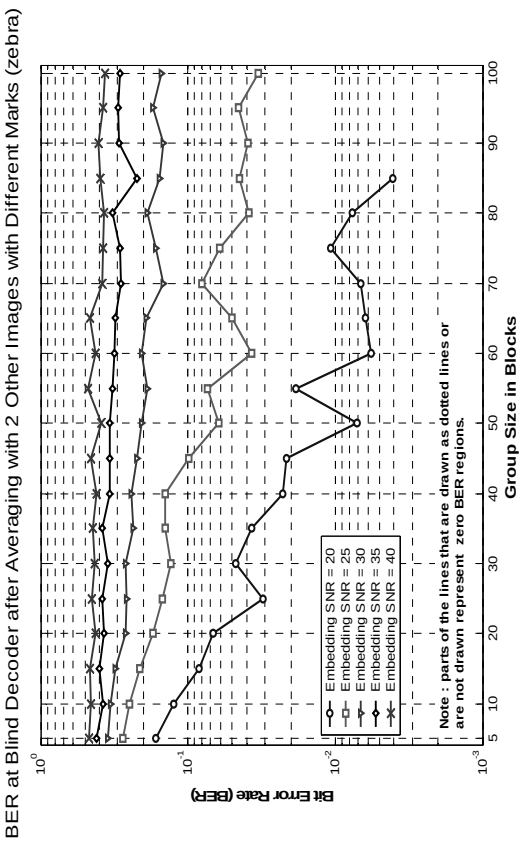


Fig. 3.19 (n)

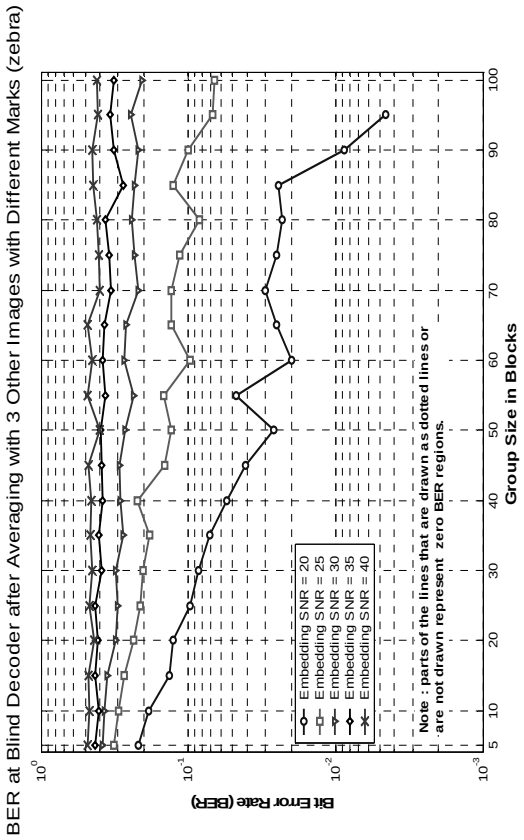


Fig. 3.19 (o)

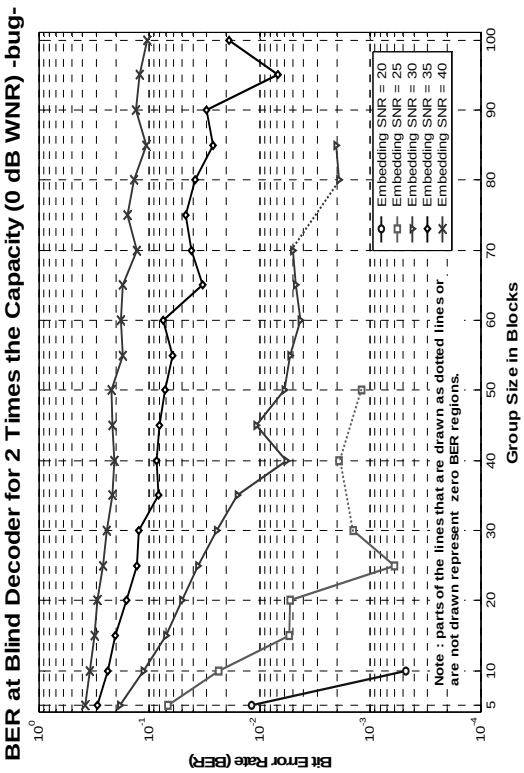


Fig. 3.20 (a)

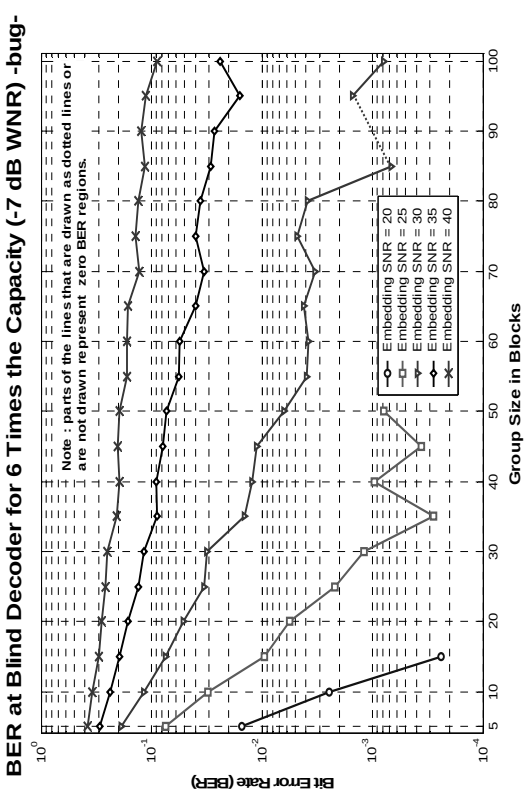


Fig. 3.20 (b)

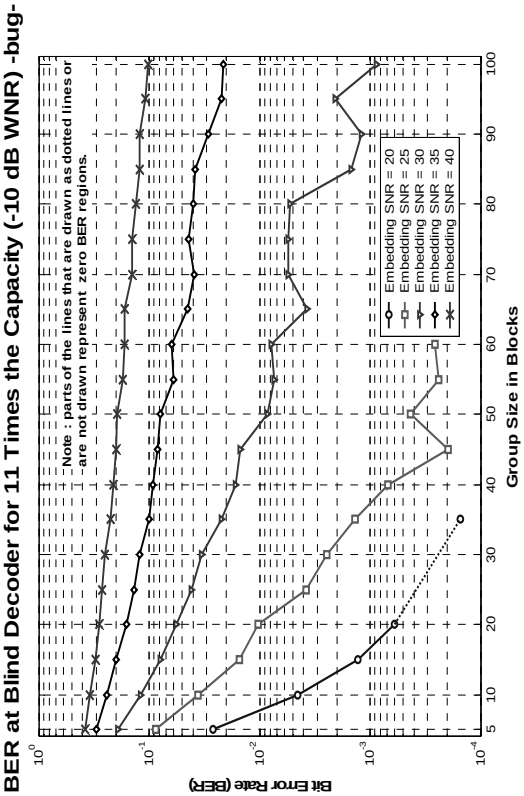


Fig. 3.20 (c)

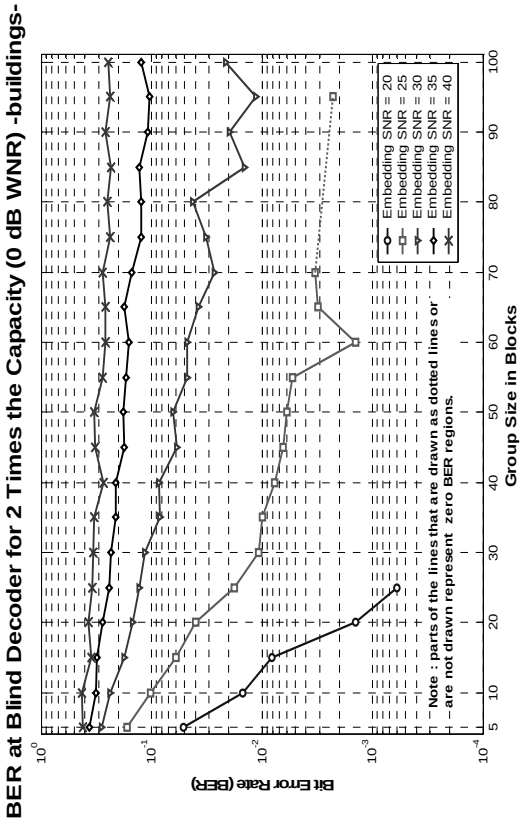


Fig. 3.20 (d)

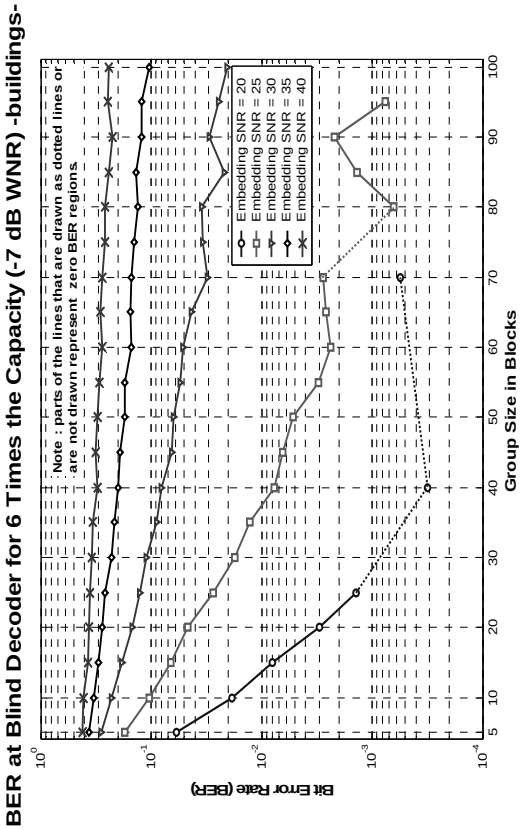


Fig. 3.20 (e)

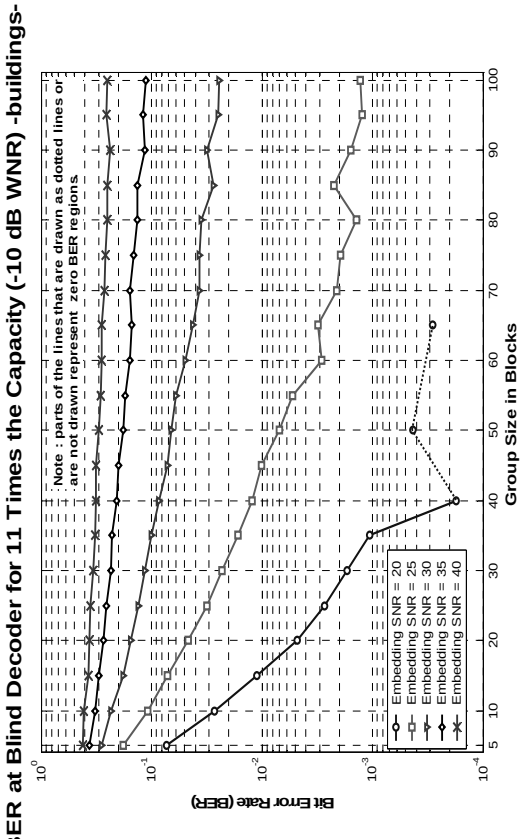


Fig. 3.20 (f)

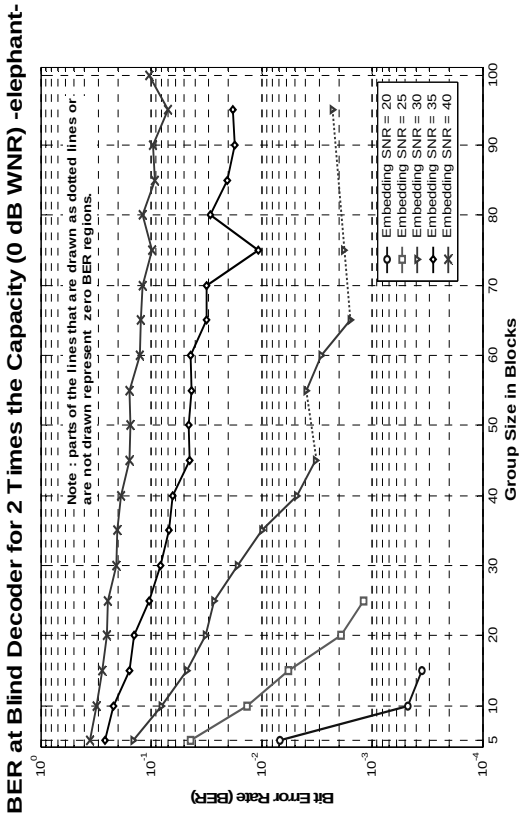


Fig. 3.20 (g)

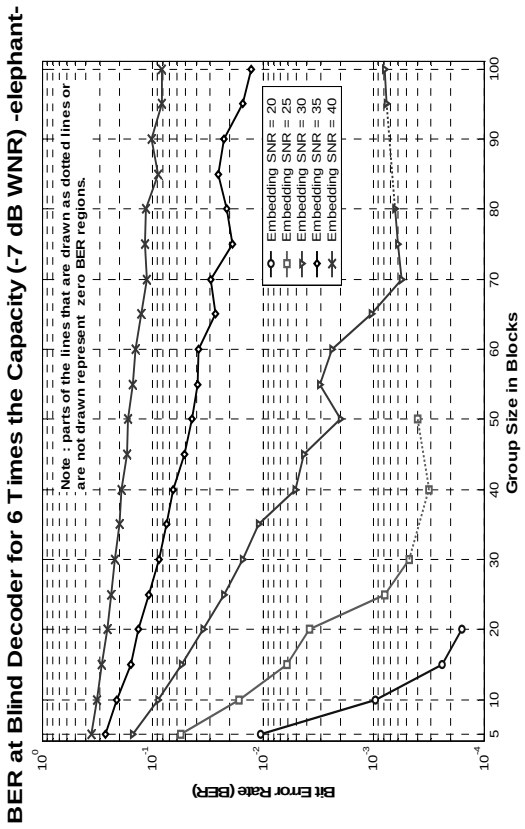


Fig. 3.20 (h)

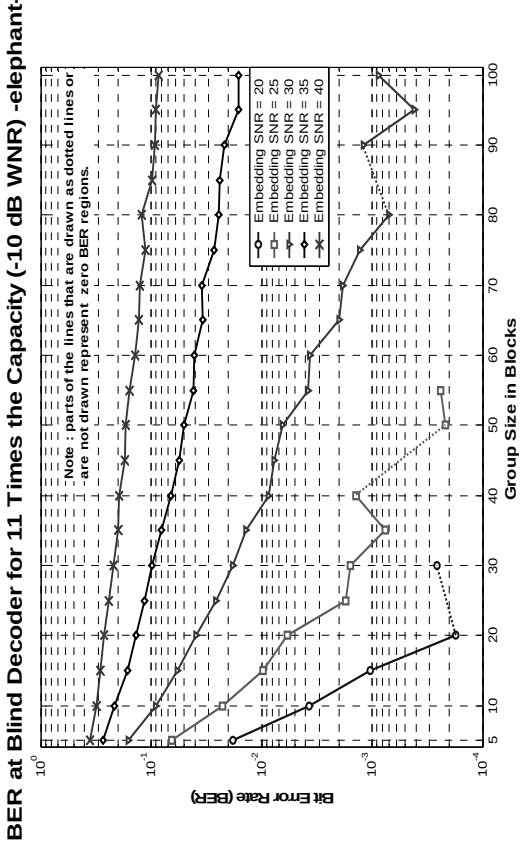


Fig. 3.20 (i)

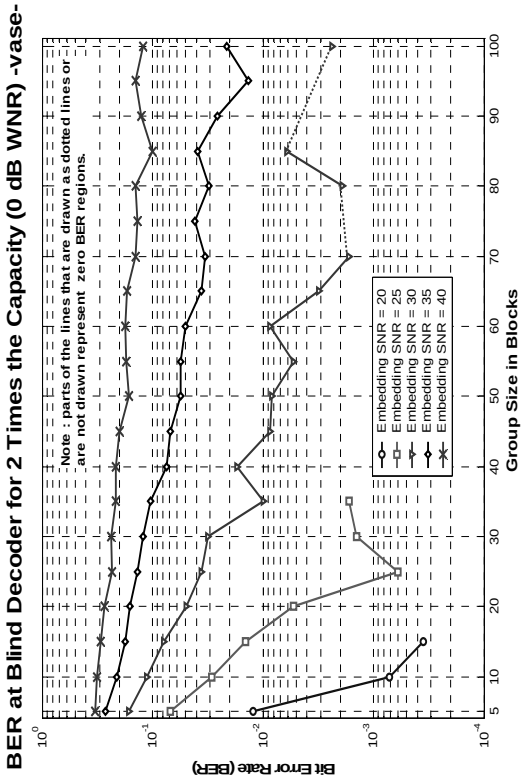


Fig. 3.20 (j)

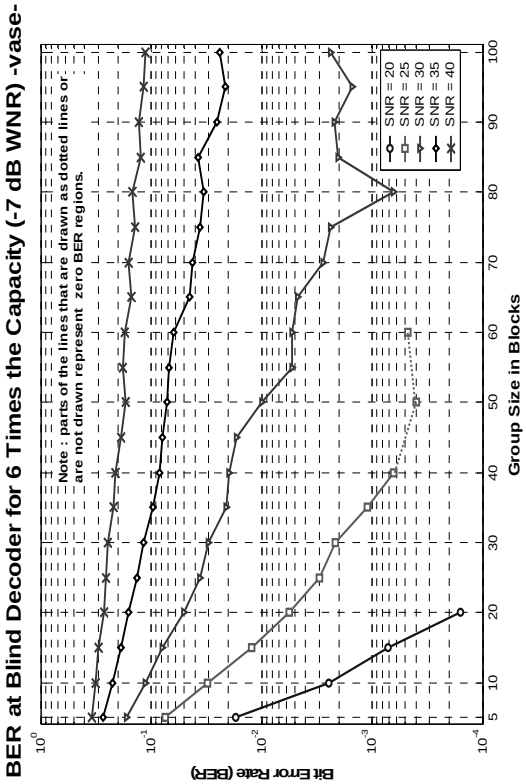


Fig. 3.20 (k)

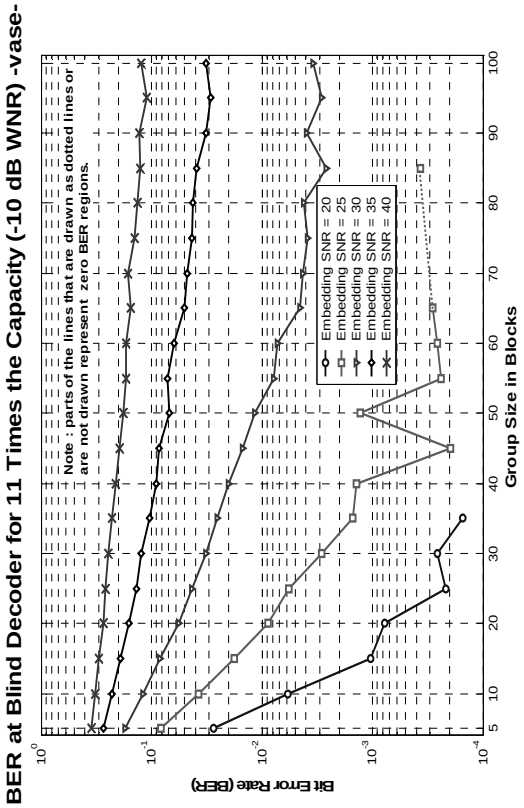


Fig. 3.20 (l)

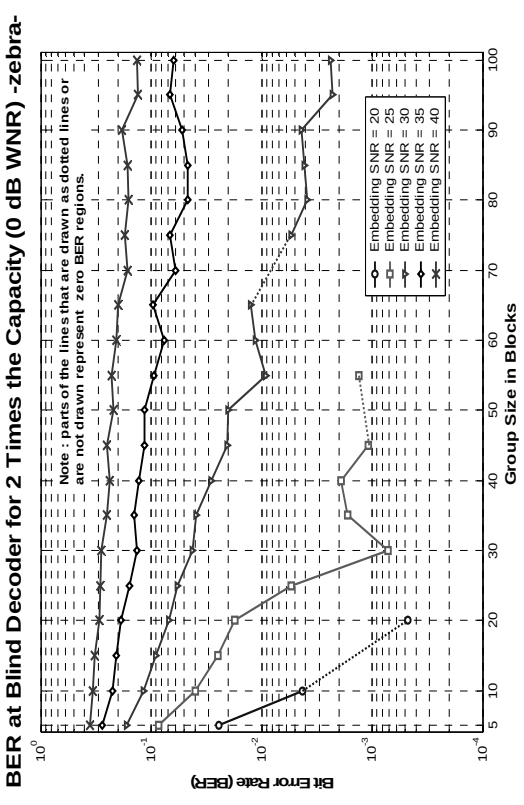


Fig. 3.20 (m)

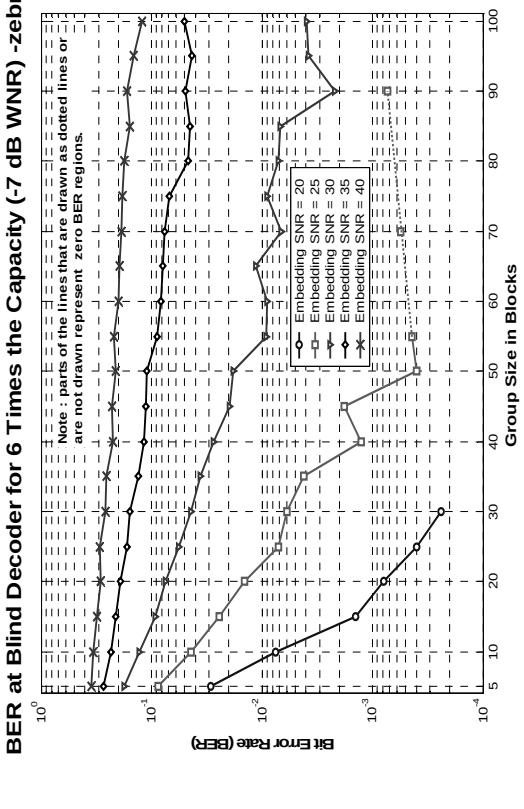


Fig. 3.20 (n)

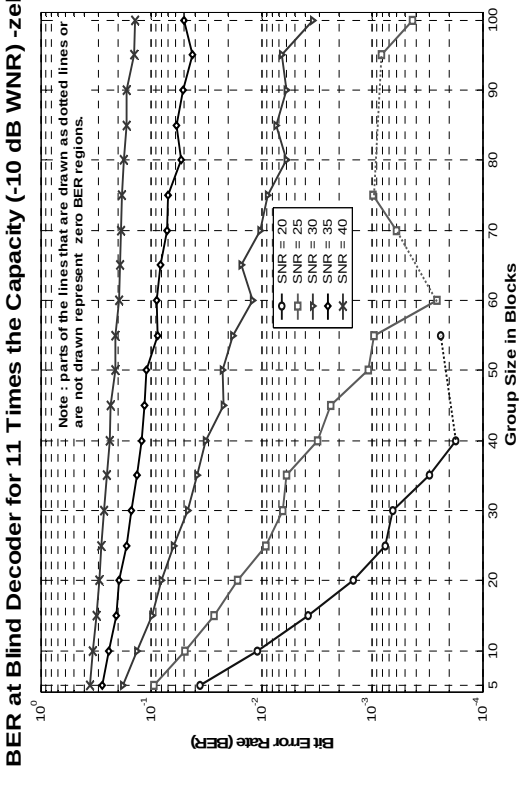


Fig. 3.20 (o)

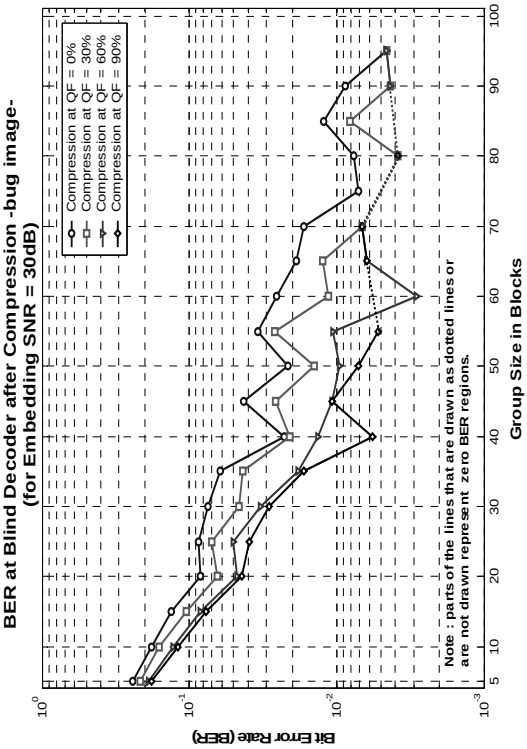


Fig. 3.21 (a)

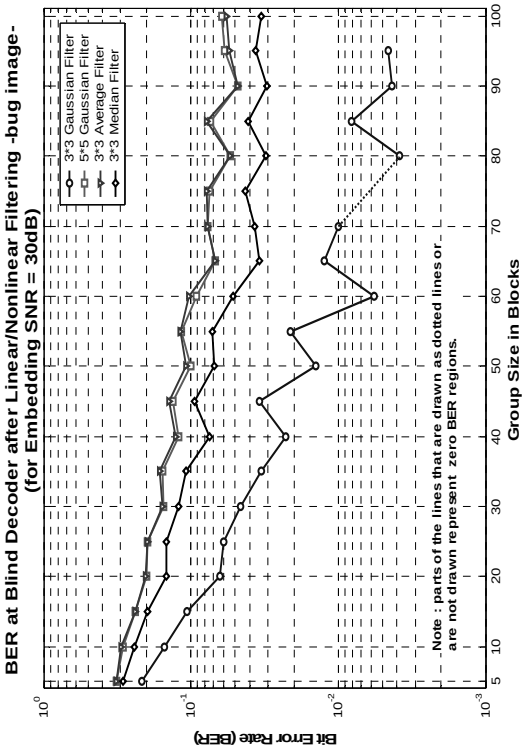


Fig. 3.21 (b)

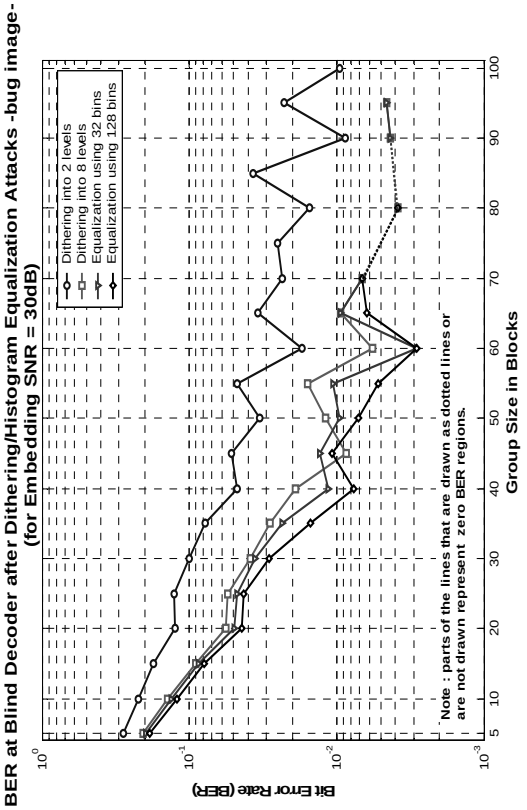


Fig. 3.21 (c)

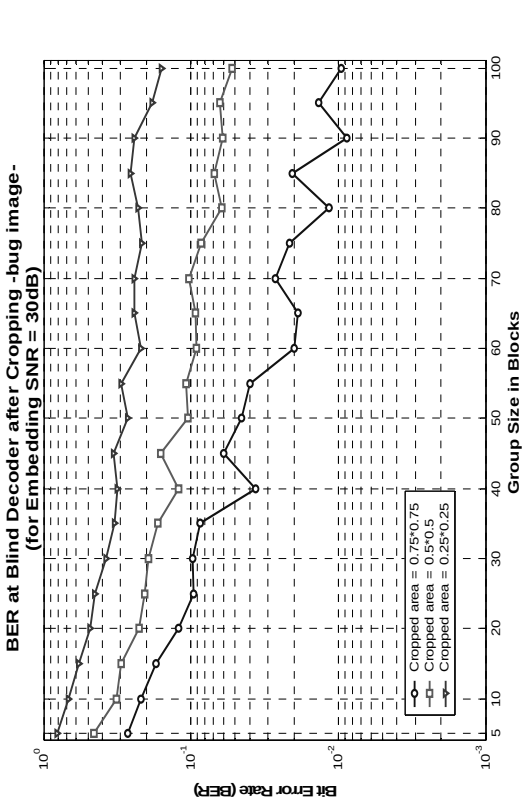


Fig. 3.21 (d)

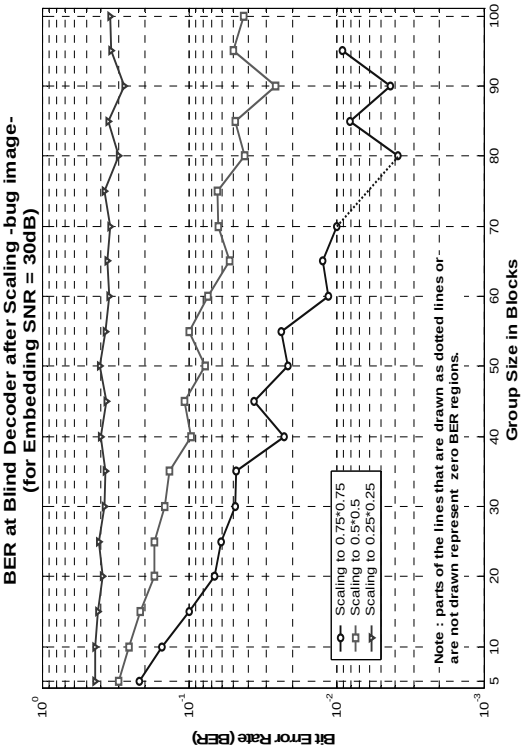


Fig. 3.21 (e)

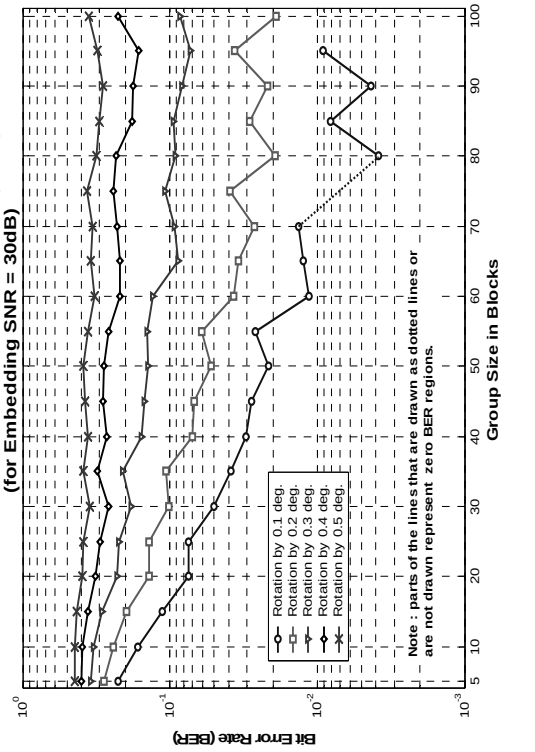


Fig. 3.21 (f)

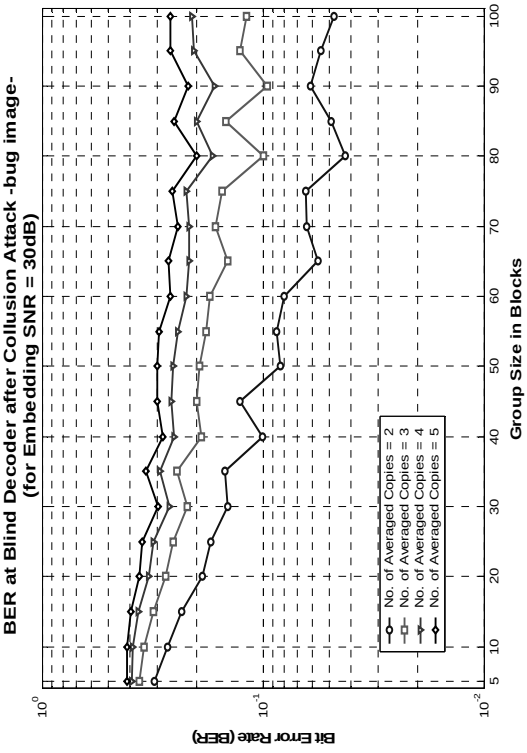


Fig. 3.21 (g)

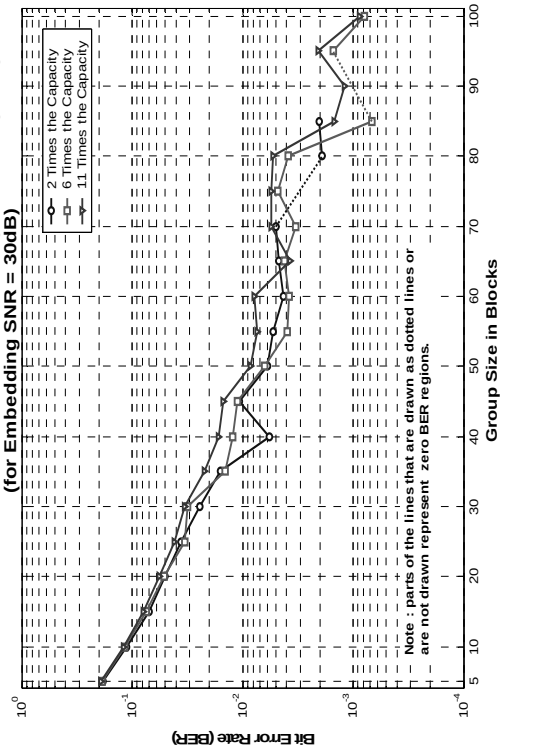


Fig. 3.21 (h)

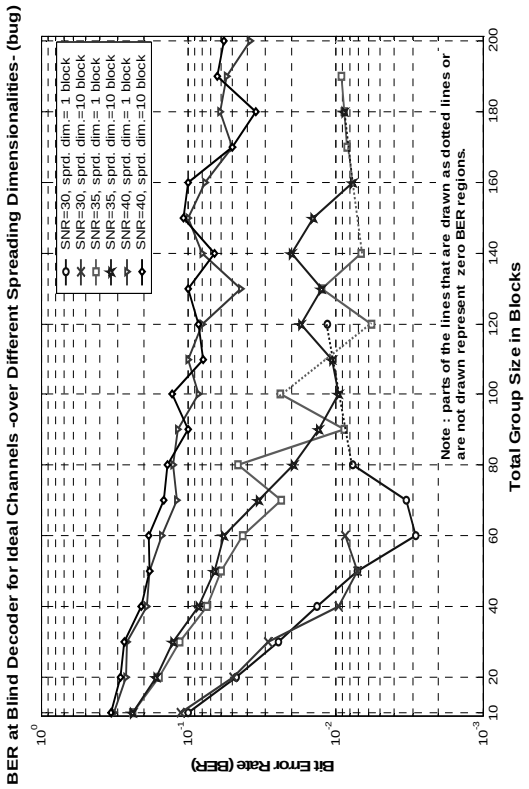


Fig. 3.22 (a)

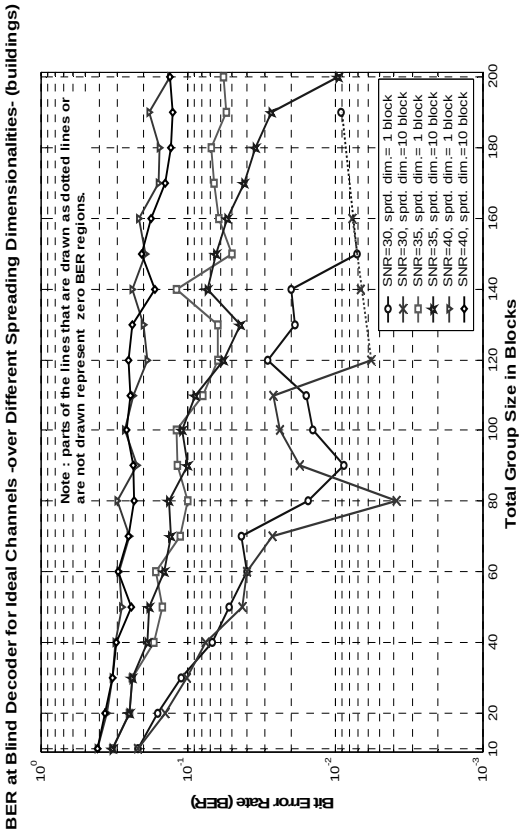


Fig. 3.22 (b)

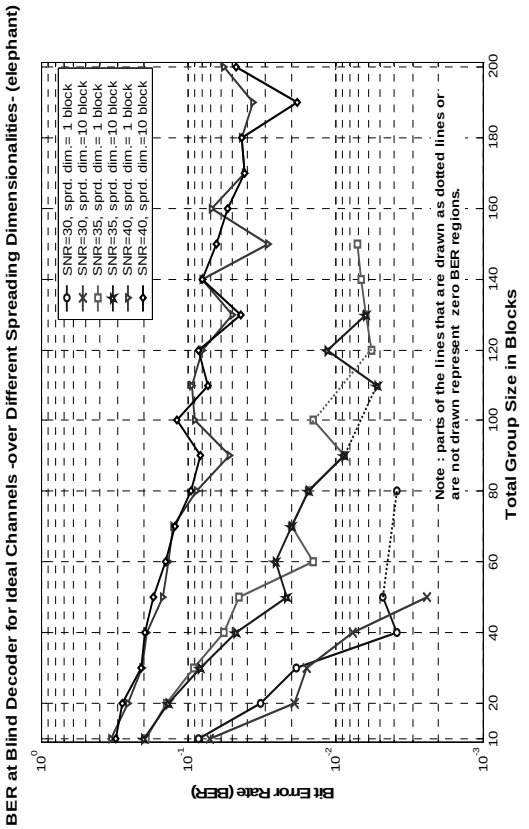


Fig. 3.22 (c)

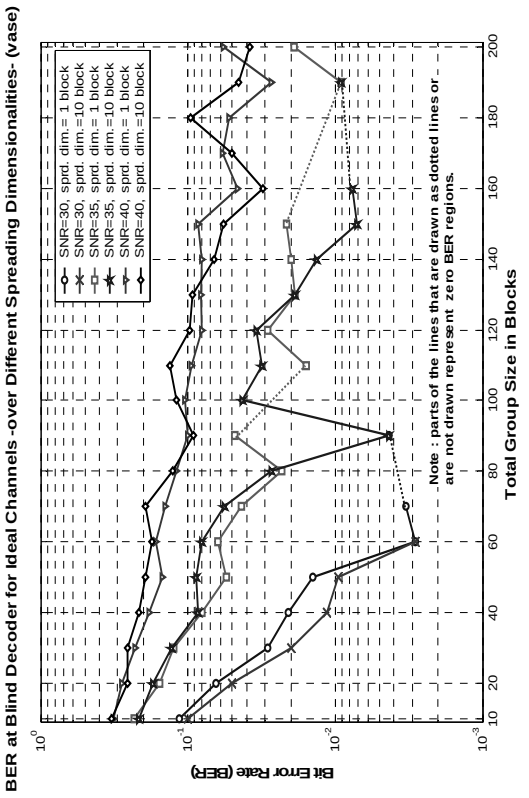


Fig. 3.22 (d)

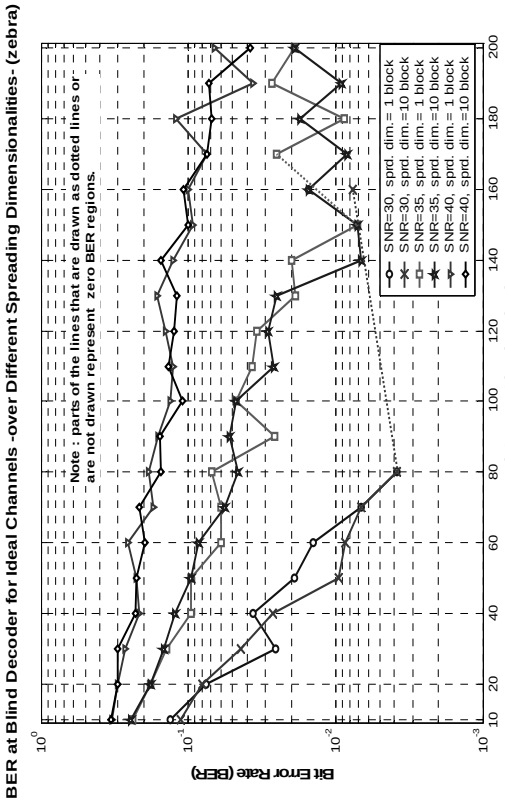


Fig. 3.22 (e)

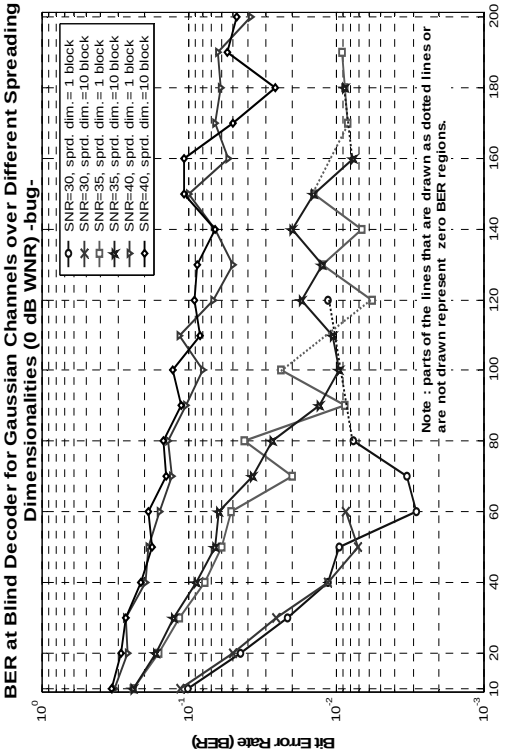


Fig. 3.23 (a)

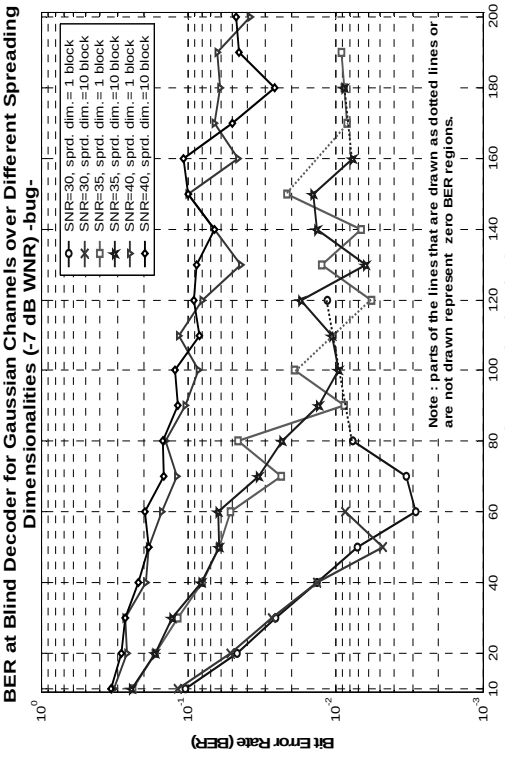


Fig. 3.23 (b)

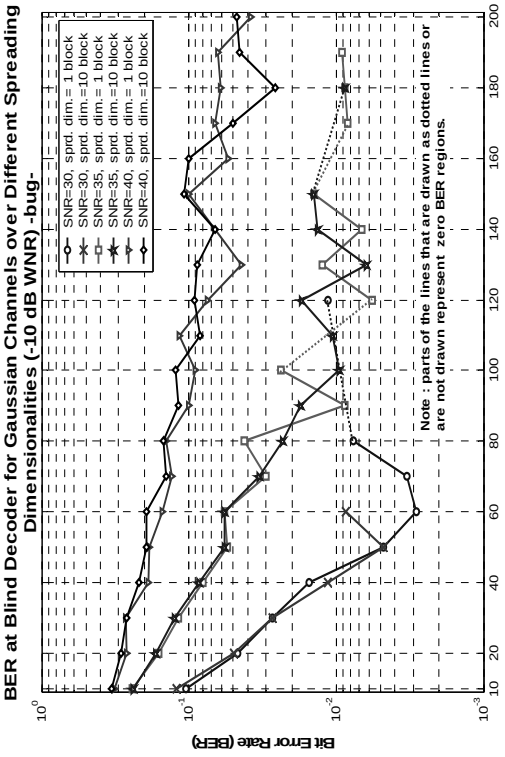


Fig. 3.23 (c)

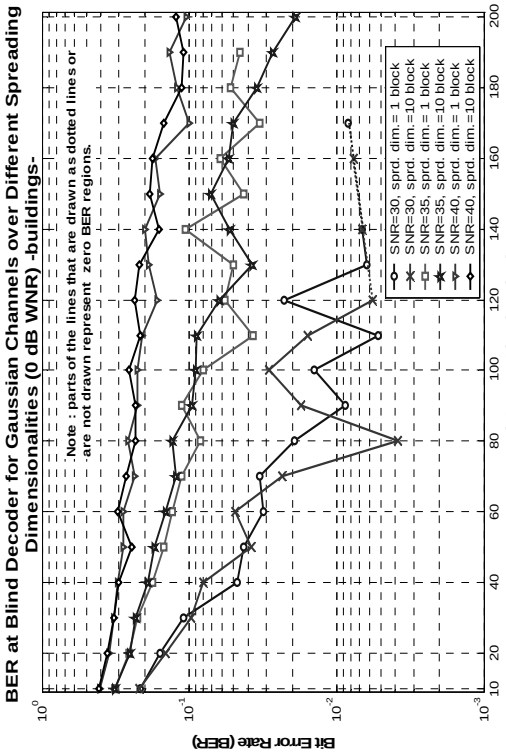


Fig. 3.23 (d)

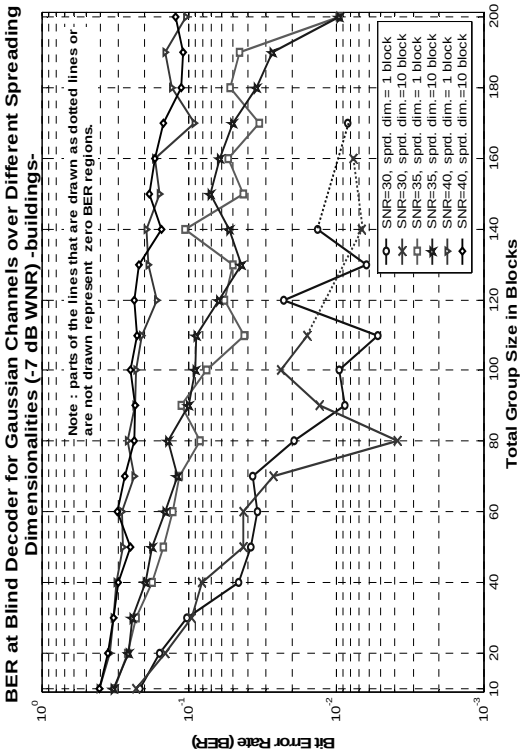


Fig. 3.23 (e)

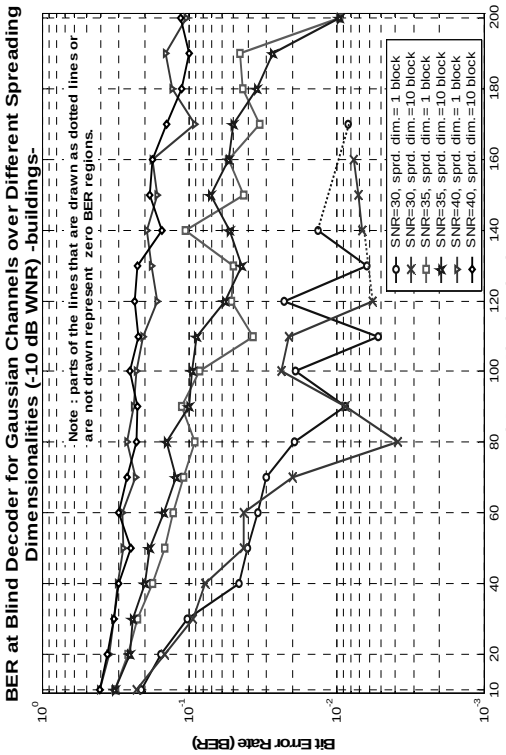


Fig. 3.23 (f)

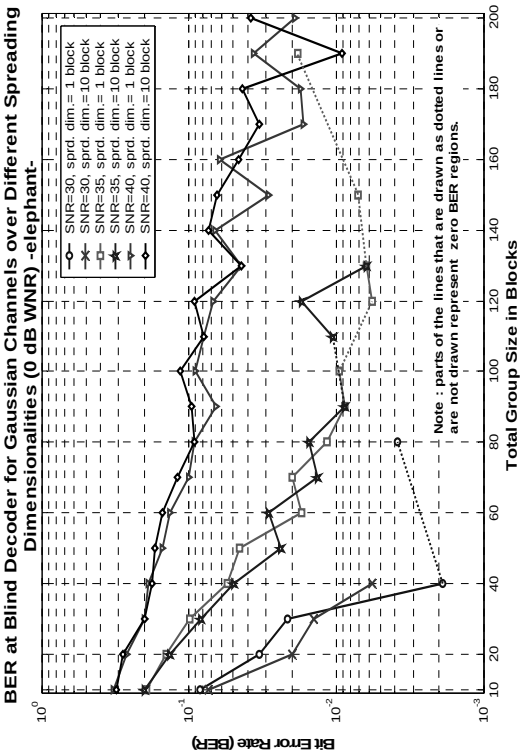


Fig. 3.23 (g)

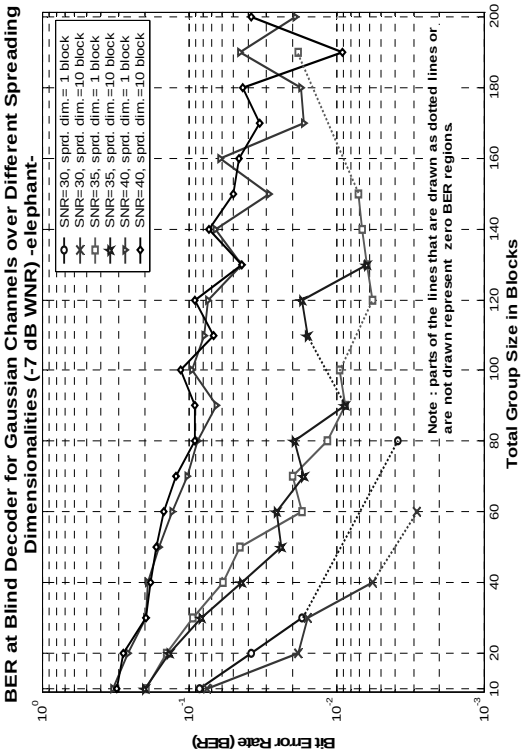


Fig. 3.23 (h)

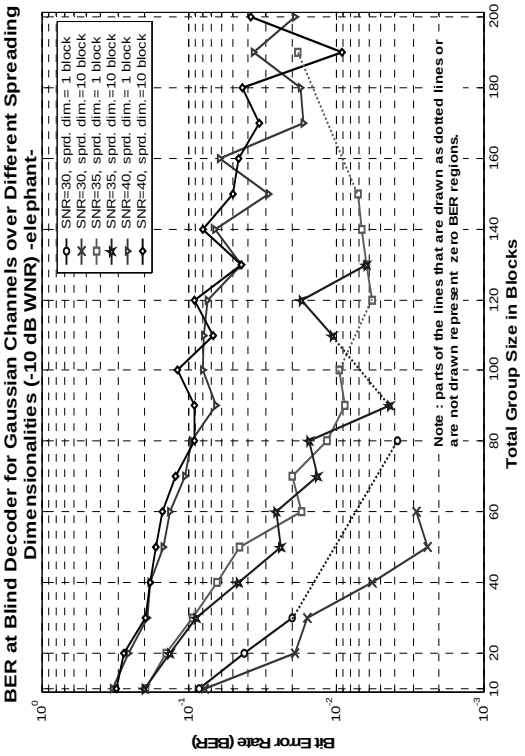


Fig. 3.23 (i)

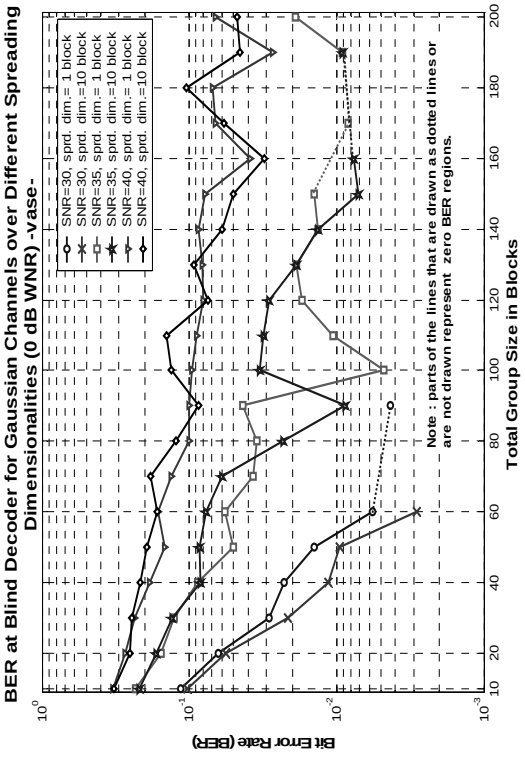


Fig. 3.23 (j)

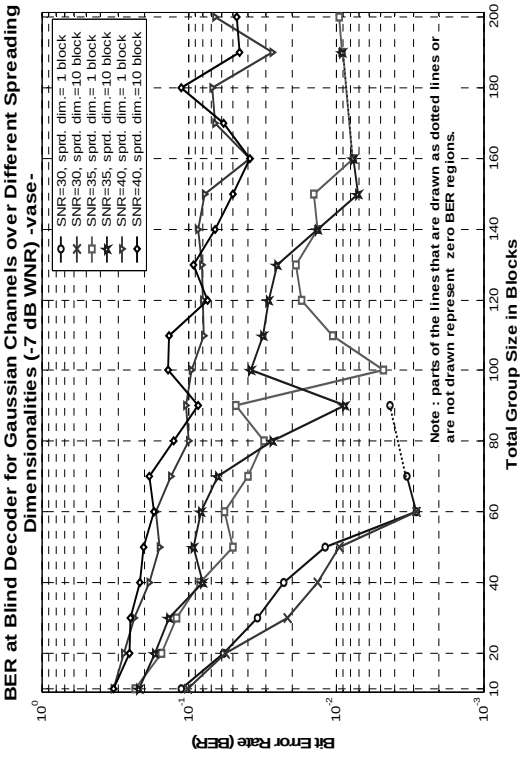


Fig. 3.23 (k)

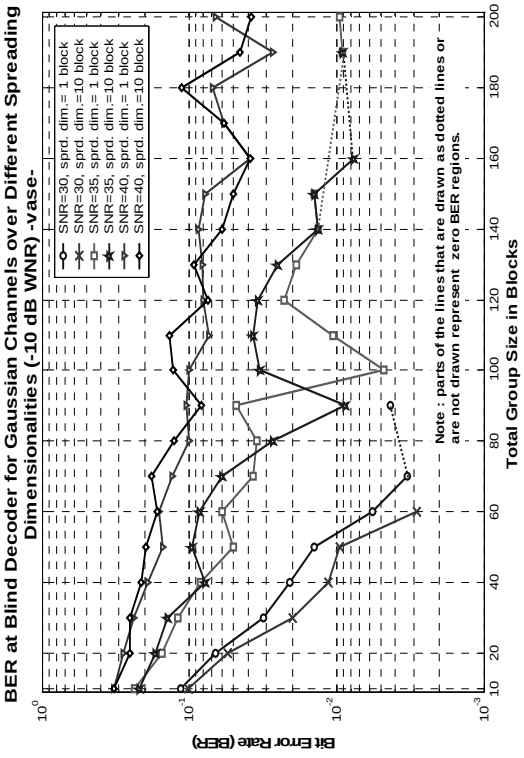


Fig. 3.23 (l)

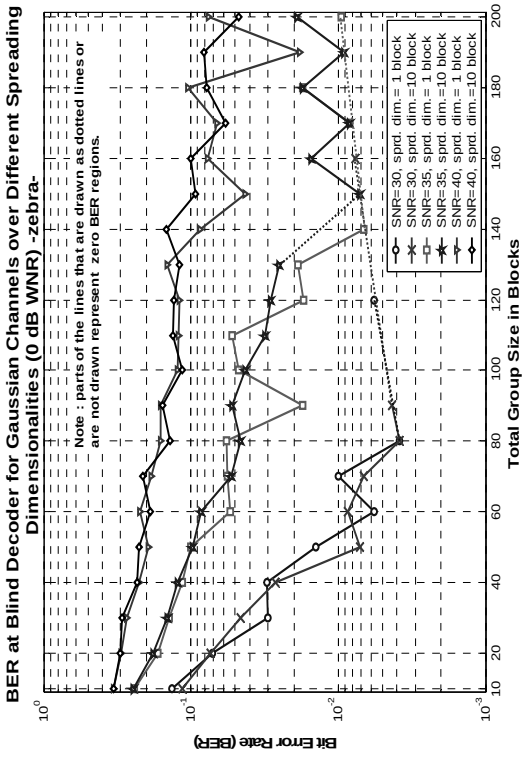


Fig. 3.23 (m)

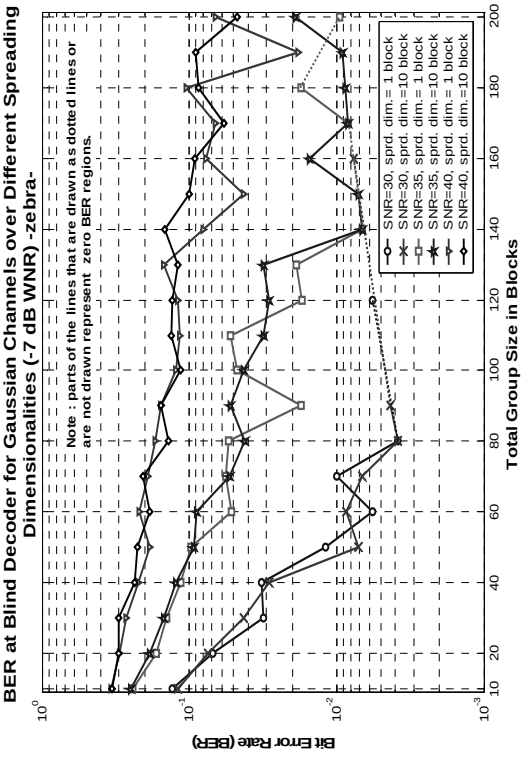


Fig. 3.23 (n)

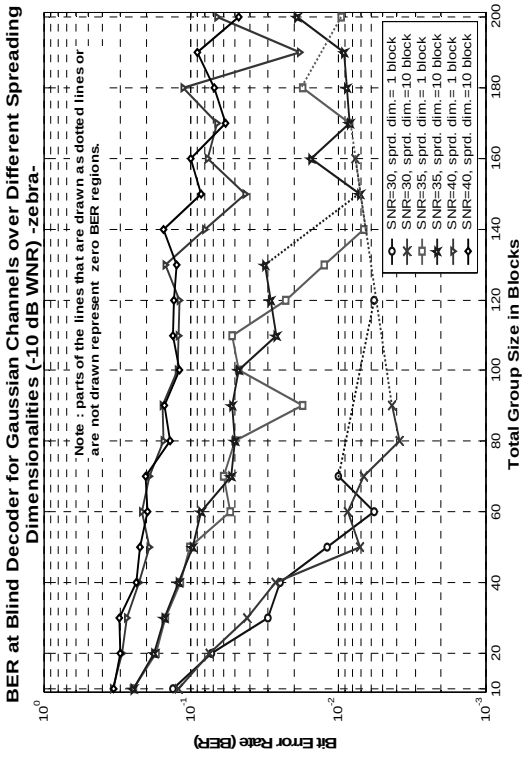


Fig. 3.23 (o)

CHAPTER FOUR

The Proposed Relative Host-Adaptation Scheme

4.1 Introduction:

As it has been mentioned earlier, host-adaptability can generally be defined as the ability of the embedding system to vary the embedded data according to the local properties of the underlying host signal [1]. Although -theoretically- this may refer to any processing that involves varying the data to be embedded according to some *non-global* host-supported information, in practice, host-adaptation has always been related to the problem of reducing the *perceptual* impact of the distortion induced by the embedded data through taking advantage of the human perception *imperfectnesses*[*]. In the next section, a *relative* host-adaptation mechanism will be proposed, that is specifically designed to take advantage of the high granularity offered by the model developed in sec. 3.3 in order to perceptually improve the system outcome without imposing any additional requirements to perfectly recover the embedded signal from adaptation distortions. Two example host-adaptation implementations will be developed and tested later to reveal the advantages of this mechanism over the traditional *absolute* (from the perspective of the ability to adapt -or scale- the individual components of the embedded signal *regardless* of the *amount* and *distribution* of the adjustments to be imposed) adaptation mechanisms.

4.2 The Proposed Relative Host-Adaptation Scheme:

The proposed host-adaptation scheme -to be described in here- is based on the watermarking model developed in the previous chapter and is completely dependent on its new implementation of redundancy and on the way this redundancy is processed.

* Remember here that the problem of searching for redundancy-significance components within host signals has been satisfactorily solved through the use of alternative signal representations, such as frequency domain representations.

It has been shown earlier that the host-perceptual-adaptation processing (see secs. 2.7 and 3.3) can functionally be partitioned into two separate steps: the host perceptual analysis step (responsible for analyzing the host signal to determine a set of the appropriate scaling arrays for reducing the perceptibility of the corresponding components constituting the corresponding watermark arrays to be embedded) and the watermark scaling process (where the produced scaling arrays are actually used to accordingly adjust the corresponding watermark arrays components). Optimally, host-perceptual-adaptation is the process of applying the most appropriate adjustment -or scale*- (given some perceptual model) to each individual component of the signal to be embedded, to accordingly *reduce* its impact on the fidelity of the underlying host signal. Such a processing though, can still severely distort the embedded data signal, making its interpretation by the decoder highly dependent on the extent to which these distortions can be equalized before decoding. This should immediately suggest that if *sufficient* side information about the host signal is available to the decoder (non-blind decoding scenario), then adaptation distortions will always be *perfectly equalizable*, given that the corresponding host-adaptation algorithm (applied by the encoder) being known to the decoder. However, since that such information about the host signal is not -generally- available to the decoder, host-adaptation schemes (at least those have been proposed till now) had to be driven far from optimality. This mainly results from the practical difficulties concerning the problem of determining a good -enough- approximation to the original version of the host signal out of the received marked -and possibly corrupted- one (the general practice under blind decoding

* Note here that in order to avoid distorting the embedded signal in an *irreversible* manner that would completely remove its information content (unknown to the decoder), host-adaptation processing had to be accomplished *independently of the embedded signal*. This has restricted the adaptation processing to multiplication based adjustments, with these adjustments being some function of the host signal and/or the side information only.

scenarios). These difficulties have generally bound the host perceptual details *exploitable* for adaptation purposes to those are capable of preserving some adequate level of accuracy in spite of the distortions induced (into the underlying host signal) later by both the embedding process and all the possible attack channels.

Referring to Fig. 3.2 of the general construction of the watermark estimate array, it may be noticed there that the way this estimate is interpreted depends on the *summations* (or the *averages*) of the elements lying along the columns of the array rather than the values of the individual elements themselves. This means that the actual values of the elements constituting these columns may still *deviate* from their designated values (exemplified by the magnified signal graph denoted by $z_2^{N/D}$ in Fig. 3.2) without disturbing the decoder output, as long as their corresponding summations (or averages) remain the same. The latter condition, however, suggests that the adjustment (deviation) to be applied to any individual element has to be *related* to the adjustments applied to the remaining elements within the same column in the watermarking array through the relationship of *zero total deviation* (that is, the summation of deviations to be applied to the elements of each column of any watermarking array should be zero), if the decoder interpretation of the corresponding watermark array is to be preserved. It is this simple idea that forms the basis of the proposed modifications to the traditional adaptation scheme.

In what follows, a simple mechanism will be devised, that is aimed at ensuring the compliance of *any* scaling array (which is generated by *any* perceptual analysis computation) with the zero total deviation relationship -mentioned above- to eliminate the need for any adaptation distortion equalization processing, on the expense of the *relativity* that this mechanism imposes on the imperceptibility scales generated.

Referring to the watermarking model described in sec. 3.3, let's consider the following scenario. Given some scaling array α^N (that can be represented by a similar construction to that shown in Fig. 3.4, with its constituting individual elements representing the imperceptibility scales produced by the adopted perceptual analysis computation for the corresponding watermark components, given the available local information about the host signal), the elements of each column of array α^N are to be *scaled* by the reciprocal of the mean value of the elements of that column to produce the *pre-equalized relative* imperceptibility scaling array $\tilde{\alpha}^N$, which has the *mean value* of the scales over each one of its constituting columns equal to one, or equivalently has the *summation* of the scales over these columns equal to their length (dimensionality) N/D (a necessary and sufficient condition -as is shown below- to achieve the desired goal, given the previously described scaling step):

$$\tilde{\alpha}_i^{N/D} = \alpha_i^{N/D} \cdot \left(\frac{1}{\bar{\alpha}_i} \right), \; i = 1, \ldots D. \qquad (4.1)$$

where $\tilde{\alpha}_i^{N/D}$ and $\alpha_i^{N/D}$ represent the ith (N/D)-dimensional columns of the arrays $\tilde{\alpha}^N$ and α^N, respectively, while $\bar{\alpha}_i$ corresponds to the mean value of the ith column of α^N, given that N, D and N/D being defined as is shown in Fig. 3.4 (see sec. 3.3.2). The generated pre-equalized scaling array $\tilde{\alpha}^N$ is then to be used to scale (a component-by-component) the corresponding components within the watermarking array W^N, such that:

$$u_i^{N/D} = \tilde{\alpha}_i^{N/D} \cdot w_i^{N/D}, \; i = 1, \ldots D. \qquad (4.2)$$

where $U_i^{N/D}$ and $W_i^{N/D}$ are the ith (N/D)-dimensional columns of the adapted watermark array U^N and non-adapted watermark array W^N, respectively. This means that the adapted watermark -now- has the summation of the elements over its constituting columns equal to:

$$\sum_{j=1}^{N/D} u_i^{N/D}(j) = \sum_{j=1}^{N/D} \tilde{\alpha}_i^{N/D}(j) \cdot w_i^{N/D}(j) = w_i^{N/D}(*) \cdot \sum_{j=1}^{N/D} \tilde{\alpha}_i^{N/D}(j)$$

$$= w_i^{N/D}(*) \cdot \frac{N}{D}, \qquad i = 1, \ldots D. \tag{4.3}$$

which is a similar result to that would be achieved without adaptation (as a result of accumulating the N/D exact repetitions of the corresponding elements of the watermarking array columns), where the index j here runs through the elements of the columns of the corresponding arrays. Note here that $W_i^{N/D}(j)$ has been factored out of the summation term, since that the non-adapted watermark arrays assume fixed values over the elements constituting their columns. These fixed values have been denoted by $W_i^{N/D}(*)$, which represents *any* element over the (N/D)-dimensional ith column of the array W^N.

It is important to note here that unlike the traditional host-adaptation scheme that generally aims at adjusting (linearly scaling) the components of the watermarking signal -to be embedded- according to the corresponding scales -generated by the adopted perceptual analysis computation- in order to *reduce* the distortion induced by the corresponding watermark components, the proposed host-adaptation scheme here adjusts the components of the watermarking array columns in order to *redistribute the perceptual distortion* induced by these columns over the underlying host components according to their *relative-perceptual-insensitivities* (the perceptual insensitivities of these host components to the induced modifications -the addition of the corresponding watermark components here- in relation to the insensitivities of the other components to be modified within the group of components on the corresponding array columns), keeping the total *signal strength* over the columns of the watermarking arrays *conserved*. This can be described as the *column-wise* (in terms of host coefficients arrays

columns) *optimal robustness-conserving* solution to the host-adaptation problem, as it aims at distributing the embedding induced perceptual distortion *uniformly* through the latter columns. Note that a better distribution (hence, a better overall fidelity) can be achieved here by increasing the size of the host coefficients arrays columns (the integrate-and-dump encoding section) towards the total number of host blocks (hence, reducing the spreading encoding section to its minimum), so that the induced distortion is more uniformly distributed over the entire host signal, while decreasing the size of these columns towards traditional spread spectrum would degrade the scheme towards nonadaptivity.

However, it should be mentioned here that although the proposed adaptation scheme can significantly improve the perceptual quality of the system outcome, yet, in order to conserve the robustness of the embedded watermarking signal it would also *inadvertently* increase the power of that signal (in a mean squared sense). This actually due to the additional AC component inserted into the columns of the watermarking arrays as a result to shifting the values of the elements of these columns from their corresponding fixed values (which would generally increase the overall variance of the embedded signal). It can also be shown that this extra AC component is to be completely *filtered out* by the accumulation process on the receiving end, which makes it *useless* for data embedding.

The proposed modifications above induce two important advantages. *First* -and most important-, is that the host-adaptation process here does no longer induce any distortion into the embedded data signal, a similar overall result to that can be achieved when sufficient information about the host signal is available to the decoder (where perfect recovery from adaptation distortions becomes possible -given the knowledge of the decoder about the specific adaptation algorithm being applied by the encoder-), except that in the proposed scenario no information about the

host signal needs to be available to the decoder, since that the adaptation distortions are being equalized *prior* to embedding now. *Second*, is that the decoder does no longer need to know anything about the adaptation processing being applied to the embedded data (nor even whether any adaptation process has taken place or not). This enables the corresponding encoder to select any adaptation algorithm (and even adopt any perceptual model) that is seen to produce a higher perceptual quality just prior to the transmission process (releasing the marked work to the public) without worrying about the ability of the corresponding decoder to perfectly recover the embedded signal (in regard to the distortion induced by adaptation processing), a result that could not have been achieved even with the private marking scenario (when the complete host signal is available to the decoder), where it has been necessary for the decoder -there- to know about the specific adaptation algorithm that has been applied to the embedded signal to be able to recover from.

4.3 Experimental Results:

In this section two very simple perceptual analysis computations will be suggested and *attached* to the host-adaptation scheme described above, so that it would be possible to examine the effect of this scheme on the experimental results achieved in the previous chapter. It is important to mention here that the computations to be developed next are not meant to be comprehensive in terms of modeling (meeting) the human perception imperfectnesses, but rather, to show the new possibilities offered by the proposed adaptation scheme.

4.3.1 Noise-masking Based Perceptual Analysis:

The first suggested host perceptual analysis method here is based on the well-known *noise-masking* (*activity-masking*) phenomenon [32, 33].

Generally, *visual masking* can be defined as any *interference* between two or more visual signals that would affect (usually decrease) their visibility [5, 33]. For example, image regions with tense and/or rapid intensity variations can -generally- mask (*perceptually hide*) signals of much greater amplitudes than those can be masked by smooth (near constant intensity) regions. This elementary observation has drawn our attention to the possibilities those can be offered by such a phenomenon to perceptually enhance the outcome of the proposed system with respect to the watermarking signals to be embedded (or *hidden*). Noise-masking (a one of the interesting visual masking phenomena) is concerned with the effect of the non-regular and highly varying [33] visual signals on the visibility of the others. However, unlike the other masking phenomena, whose effect do especially rely on the shape and the orientation of the masking patterns (which would require some heavy additional processing to determine), noise-masking effects can be fairly estimated with only slight extra processing in the presence of sufficient frequency domain information, since that the noise contents of the different image regions can be well-defined in terms of the high frequency spectrum. In what follows, a simple computation will be suggested, that mainly aims at determining the noise contents of the underlying host image blocks in order to develop some estimate (represented by the generated scaling arrays) of the *relative* masking capabilities (in the relative host-adaptation sense) of these blocks, which can be used later to accordingly *redistribute* the power of the watermark to be embedded over them.

The suggested host perceptual analysis computation here is built upon the blockwise frequency domain information supported by the second stage of the watermark embedding system (described in sec. 3.3.2). Here, the noise-masking effect of each 8*8 host image block on the corresponding watermarking block to be embedded within (or more

accurately the *block-wide* portion of the corresponding watermark segment) is considered to be *proportional* (note that this assumption is made here to avoid complicating the computations) to the total *energy* of the m-highest frequency 2-dimensional DCT coefficients of these corresponding host blocks. The generated arrays of *energy indexes* are then passed to the pre-equalization processing step (described in sec. 4.2) to produce the corresponding pre-equalized scaling arrays. Note that the scales produced in here represent the noise contents (hence, -proportionally- the masking effect) of the *entire* corresponding host image blocks, and are hence to be used to scale the entire block-wide portions of the watermark segments to be embedded into these blocks. However, this computation can still be modified to deal with the individual elements constituting these blocks.

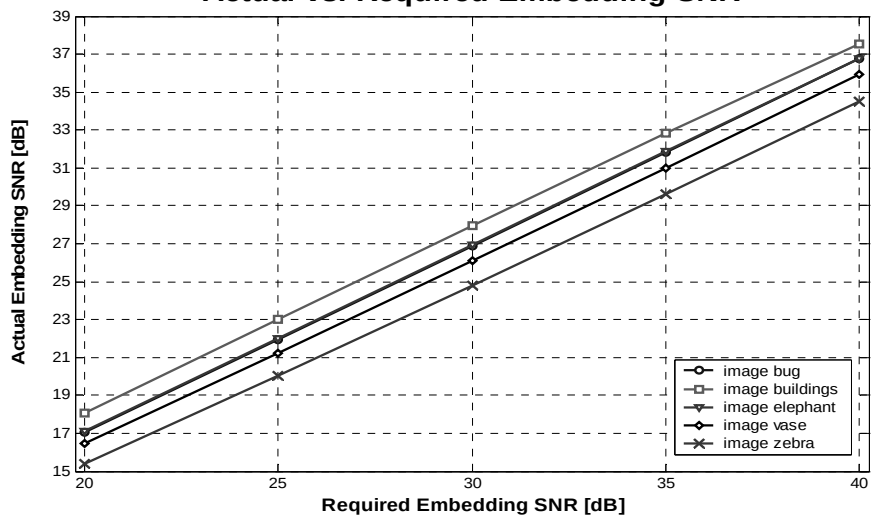
The computation above can be summarized in these two steps:

1. For each host image block, the corresponding imperceptibility scale can be found through calculating the total energy of the m-highest frequency 2-dimensional DCT coefficients.
2. The generated arrays of scales are then to be accordingly pre-equalized as is shown in equation 4.1.

For the tests performed in here, the energy of the 23-highest frequency DCT coefficients of the corresponding host image blocks is chosen to be calculated (it is practically found that this number is not very critical). The perceptual enhancements imposed by the host analysis computation described above are shown in Figs. 4.2 through 4.6. The tests performed in sec. 3.4.3 (on the nonadaptive implementation of the proposed watermarking system) have all been repeated here on the host-adaptive version of the proposed watermarking system. The corresponding results of these new tests are shown in Figs. 4.8 through 4.19. The effect of the proposed host-perceptual-adaptation process on the power of the

embedded watermarks is shown in Fig. 4.1 in terms of the resulting reduction in the embedding SNR. Note that the difference between the actual and the required embedding SNRs here depends on the power of the additional AC component inserted into the watermarking signal as a result to the adjustments applied by the adaptation process. However, this difference can still be much reduced by simply *limiting* some of the unacceptably large adjustments to be applied by the adaptation process, as will be seen in the next section.

Fig. 4.1
Actual vs. Required Embedding SNR



4.3.2 Edge-enhanced Masking Based Perceptual Analysis:

An important disadvantage to the spectral method suggested above for determining the noise contents of the host image blocks is the inability of this simple computation to distinguish regions with tense non-regular variations from those are containing rapid intensity changes (corresponding to objects *edges*), which generally have less masking capabilities than the previous (noisy) regions, yet, a relatively high energy content within the high frequency band. This content misjudgment could actually concentrate a *relatively* large share of the watermarking signal power into these regions, which would induce some undesirable visual artifacts around the objects edges within the watermarked images (as can

be noticed around the boughs of the flowers in the vase image Fig. 4.5-b and above the back and around the white and black strips of the zebra in Fig. 4.6-b). To deal with this problem, host images are suggested to be comprehensively *searched* for these blocks where such edges lie, so that their share of the watermarking signal power (actually their high frequency contents indexes) would then be *decreased* by some suitable fraction (note here that these adjustments are justified -with respect to the invariance of the total embedded signal strength- through the pre-equalization process that follows this content determination process).

To efficiently handle the *edge detection* job, Canny's algorithm (filter) is suggested to be used here. This algorithm finds edges by looking for local maxima over the gradient (calculated using the derivative of Gaussian filter) of some smoothed version of the image. However, unlike all the other edge detection algorithms, Canny's algorithm uses two thresholds to detect both strong and weak edges, with the weak edges being included only when they are connected to strong ones, which would reduce the probability of this algorithm to be fooled by noise [34]. This actually is important to prevent this step from undoing the effect of the previous noise-masking based power redistribution step (note that this would happen when the edge detection algorithm marks too many noisy image regions as containing some edges, which would cause their share of the watermarking signal power to be reduced on the expense of the other relatively more smooth regions, whose share of this power is to be increased to accordingly justify the former reduction).

The effect of this edge-enhancement step on the perceptual quality of the marked zebra image is shown in Fig 4.7-b, while the corresponding output of the Canny edge detector is shown in Fig. 4.7-a. Note here that this algorithm can deal with the visual artifacts induced around the detected edges only.

(a) Nonadaptive watermark embedding method (**bug** image).

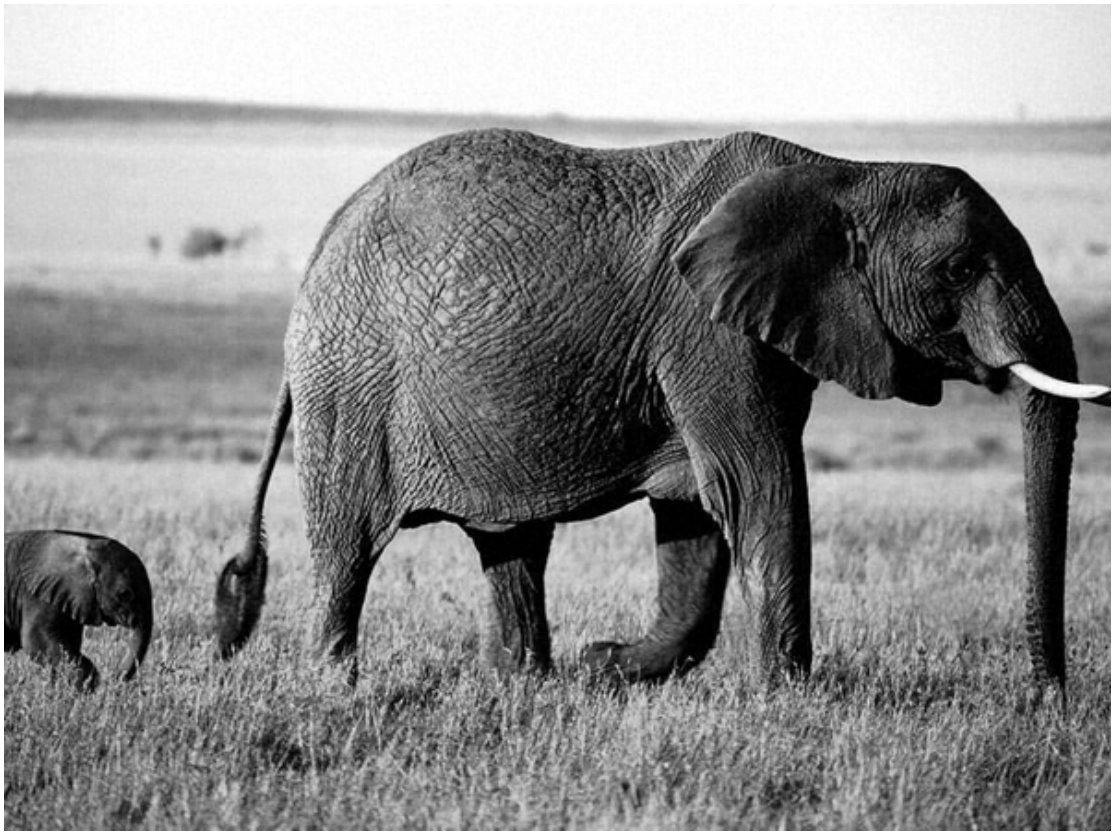
(b) Noise-masking adaptive watermark embedding method (**bug** image).

Fig. 4.2 Adaptive vs. nonadaptive watermark embedding methods. The **bug** images here are watermarked with 56 data bits, at an embedding SNR = 30 dB, and are scaled to (92% * 92%) to fit them in here.

(a) Nonadaptive watermark embedding method (**buildings** image).

(b) Noise-masking adaptive watermark embedding method (**buildings** image).

Fig. 4.3 Adaptive vs. nonadaptive watermark embedding methods. The **buildings** images here are watermarked with 56 data bits, at an embedding SNR = 30 dB, and are scaled to (92% * 92%) to fit them in here.



(a) Nonadaptive watermark embedding method (**elephant** image).

(b) Noise-masking adaptive watermark embedding method (**elephant** image).

Fig. 4.4 Adaptive vs. nonadaptive watermark embedding methods. The **elephant** images here are watermarked with 56 data bits, at an embedding SNR = 30 dB, and are scaled to (92% * 92%) to fit them in here.

(a) Nonadaptive watermark embedding method (**vase** image).

(b) Noise-masking adaptive watermark embedding method (**vase** image).

Fig. 4.5 Adaptive vs. nonadaptive watermark embedding methods. The **vase** images here are watermarked with 56 data bits, at an embedding SNR = 30 dB, and are scaled to (92% * 92%) to fit them in here.

(a) Nonadaptive watermark embedding method (**zebra** image).

(b) Noise-masking adaptive watermark embedding method (**zebra** image).

Fig. 4.6 Adaptive vs. nonadaptive watermark embedding methods. The **zebra** images here are watermarked with 56 data bits, at an embedding SNR = 30 dB, and are scaled to (92% * 92%) to fit them in here.

(a) The output of Canny edge detection algorithm (**zebra** image).

(b) Noise-masking adaptive watermark embedding method (**zebra** image).

Fig. 4.7 Edge-enhanced adaptive watermark embedding method. The image (b) here is watermarked with 56 data bits, at an embedding SNR = 30 dB. Both images are scaled to (92% * 92%) to fit them in here.

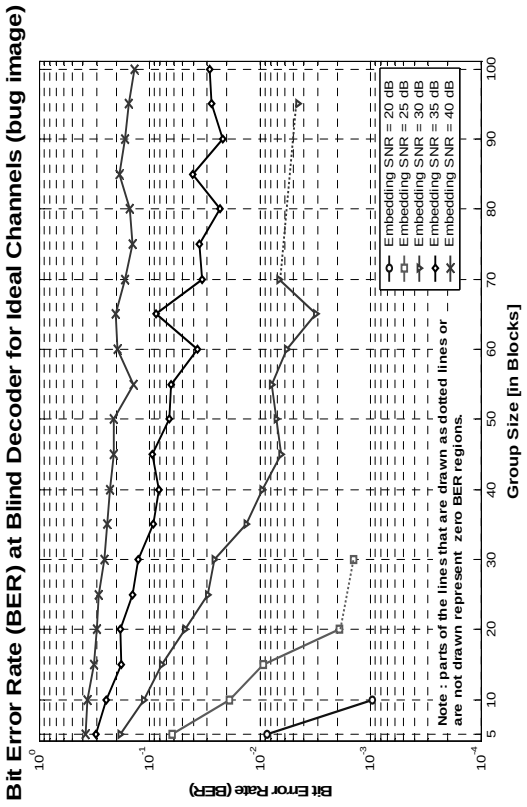


Fig. 4.8 (a)

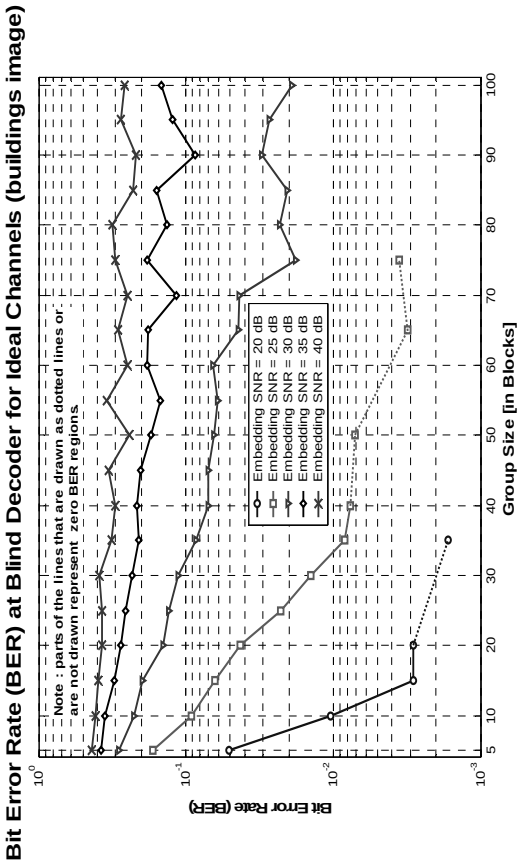


Fig. 4.8 (b)

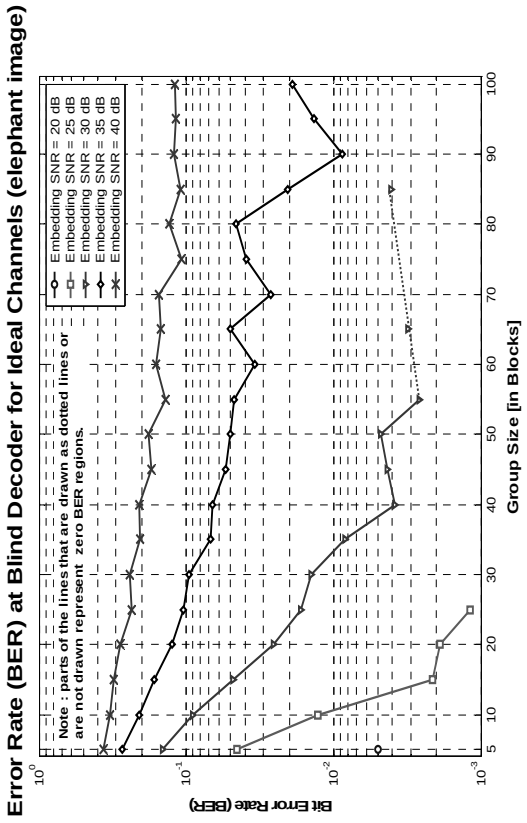


Fig. 4.8 (c)

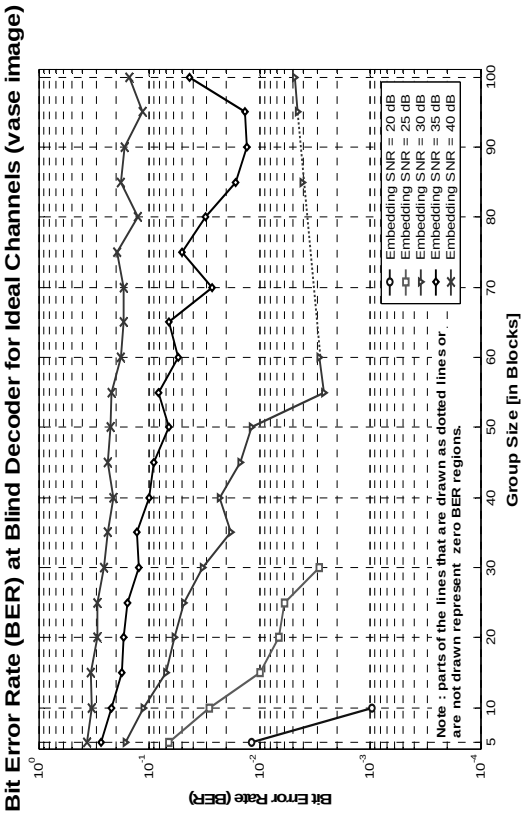


Fig. 4.8 (d)

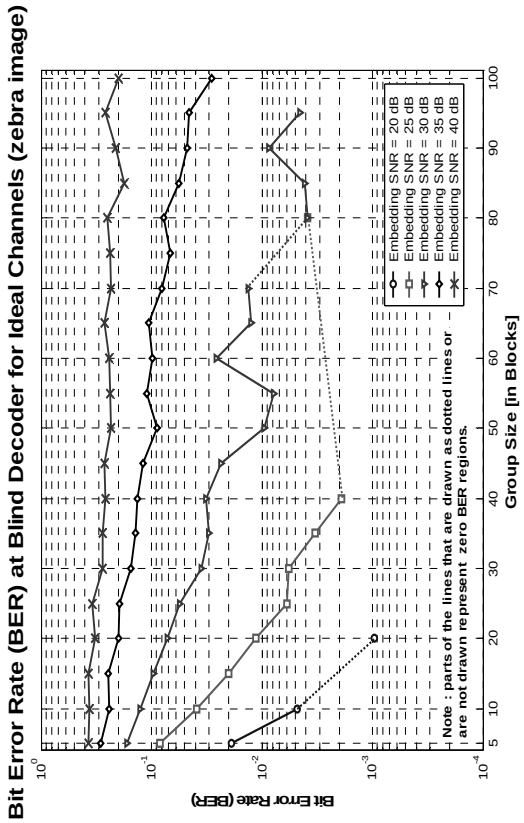


Fig. 4.8 (e)

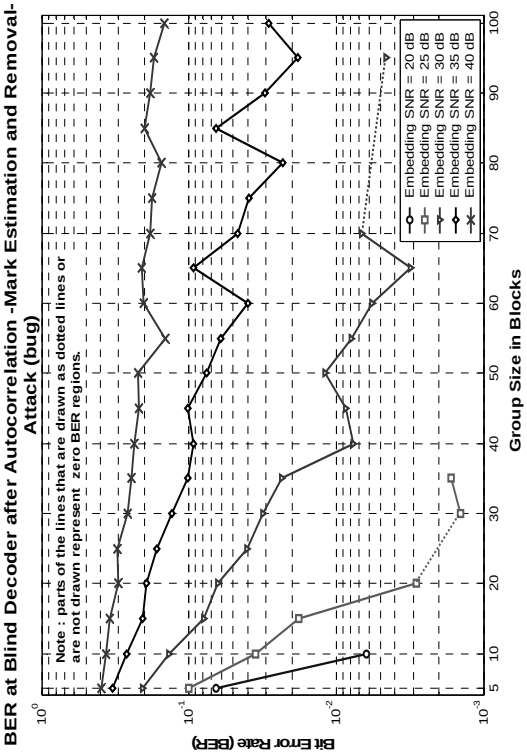


Fig. 4.9 (a)

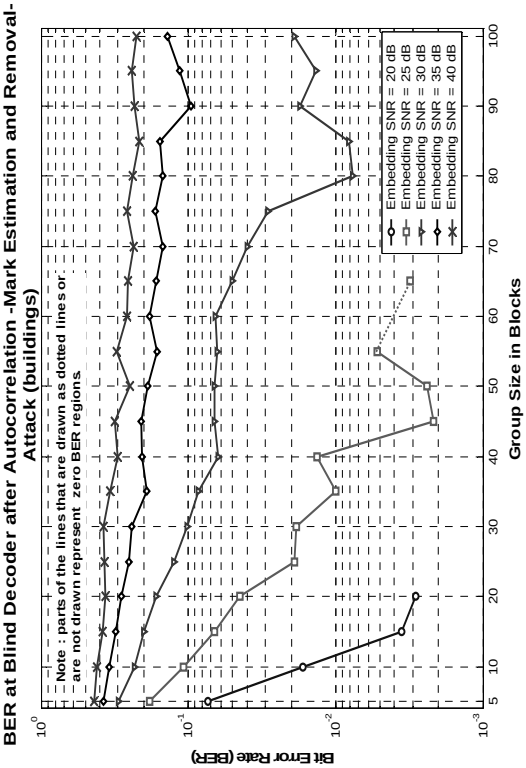


Fig. 4.9 (b)

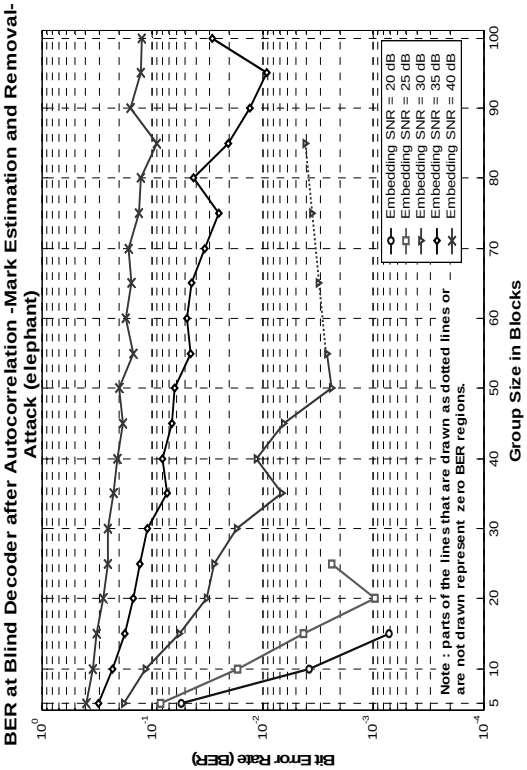


Fig. 4.9 (c)

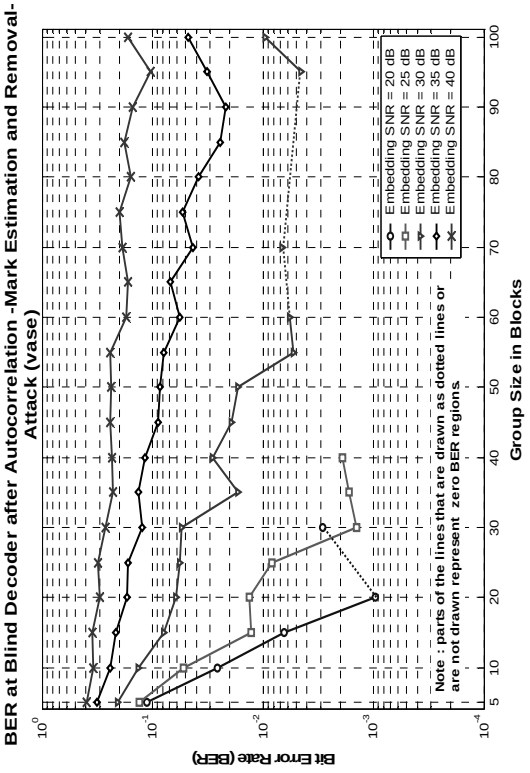


Fig. 4.9 (d)

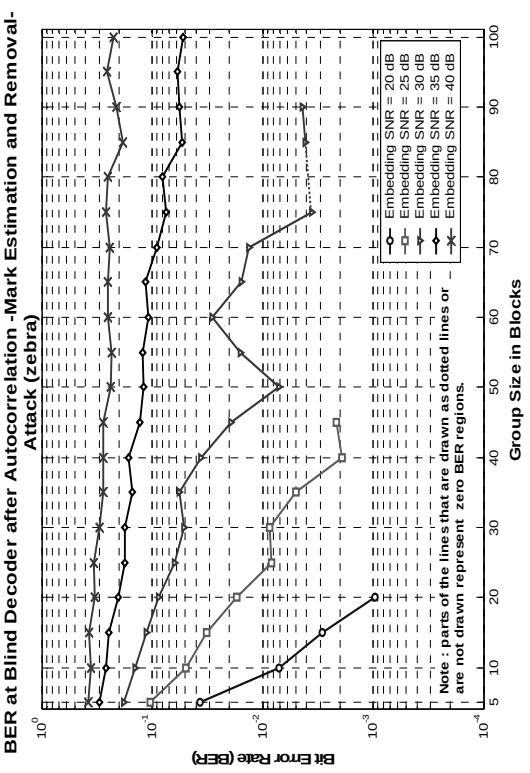


Fig. 4.9 (e)

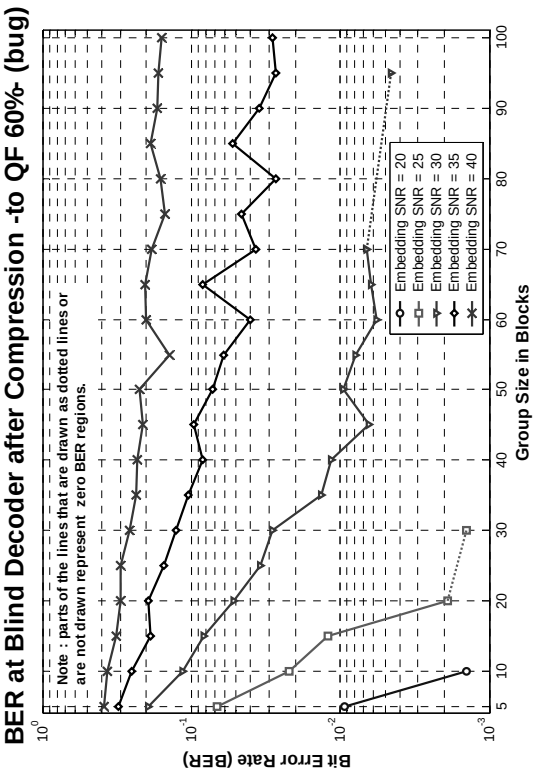


Fig. 4.10 (a)

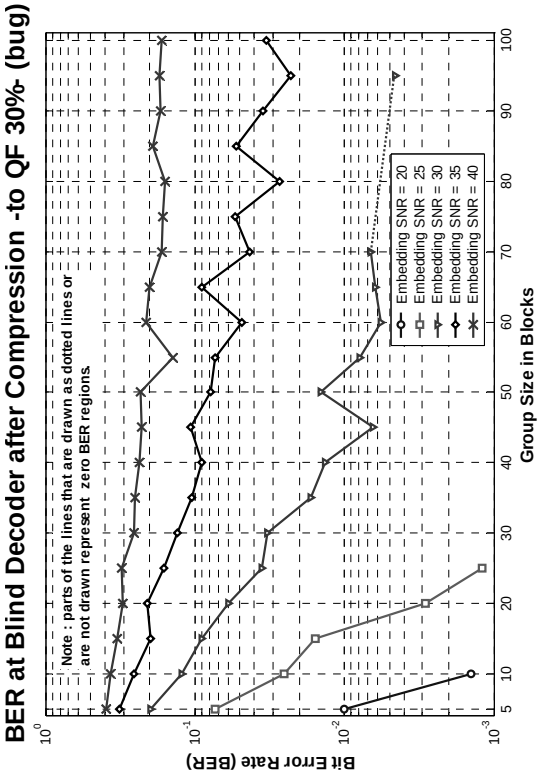


Fig. 4.10 (b)

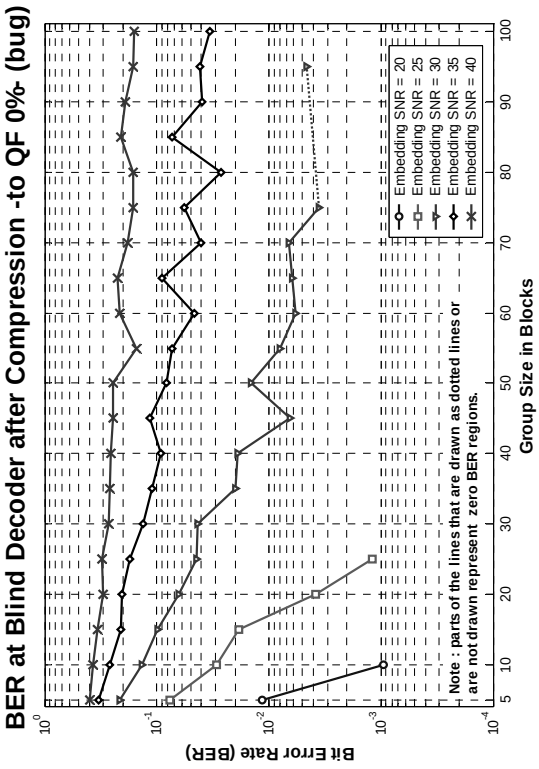


Fig. 4.10 (c)

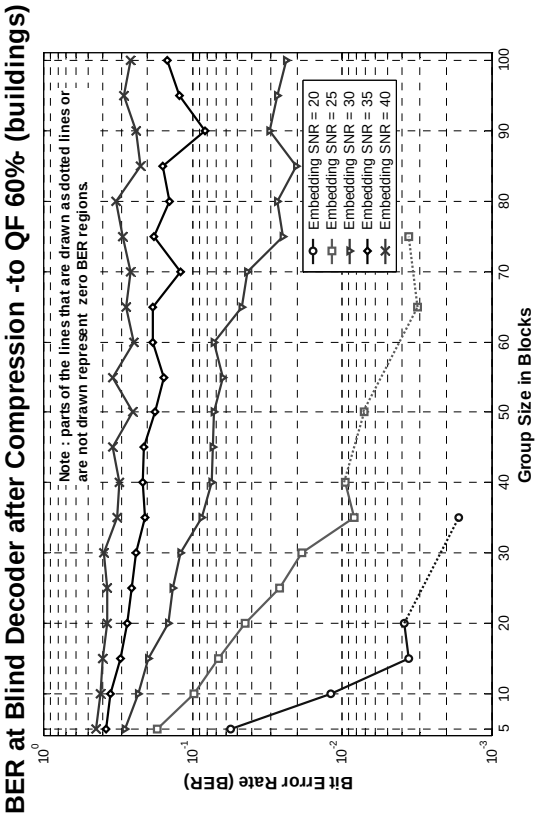


Fig. 4.10 (d)

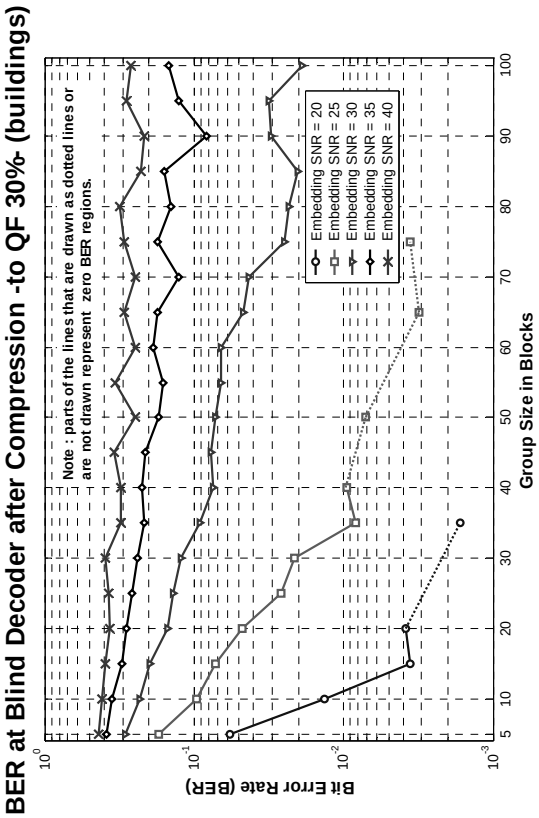


Fig. 4.10 (e)

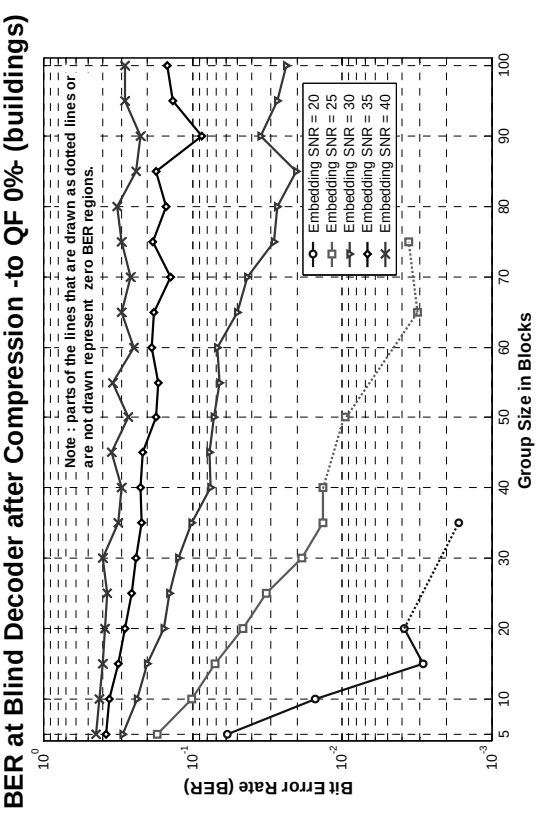


Fig. 4.10 (f)

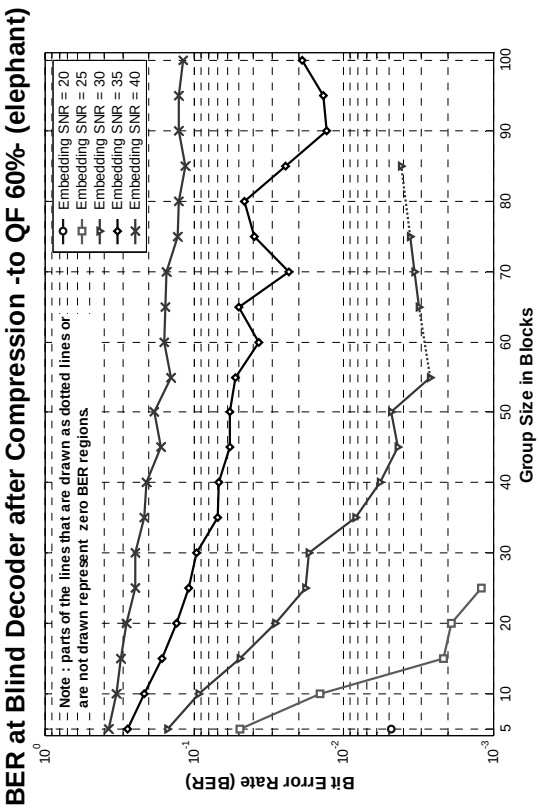


Fig. 4.10 (g)

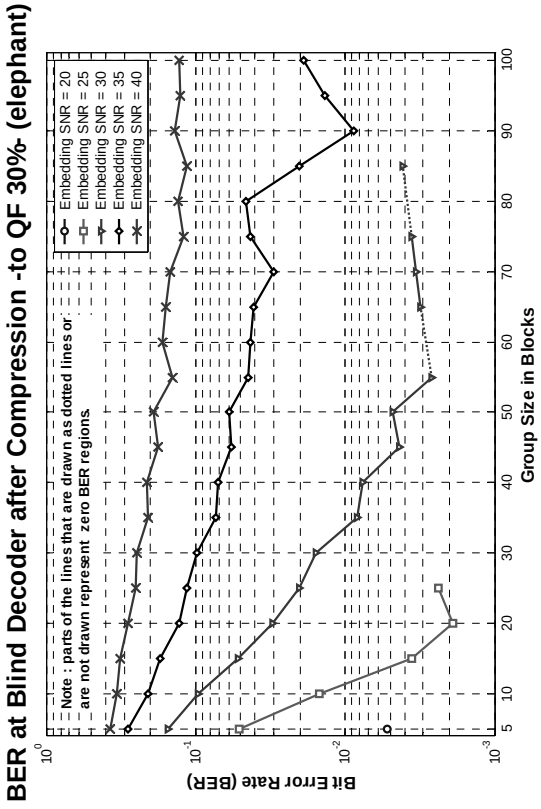


Fig. 4.10 (h)

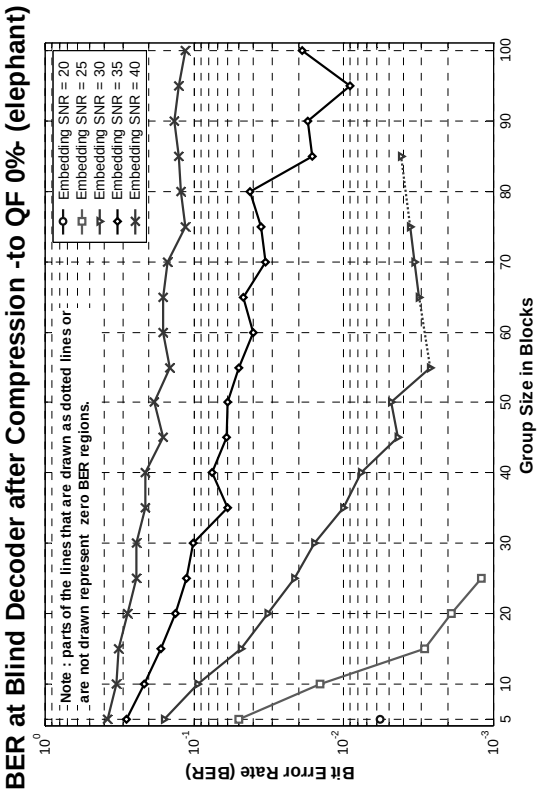


Fig. 4.10 (i)

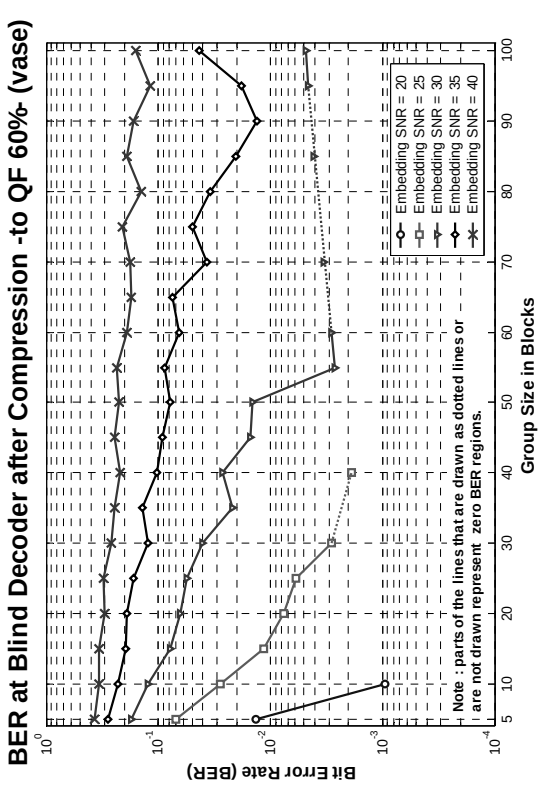


Fig. 4.10 (j)

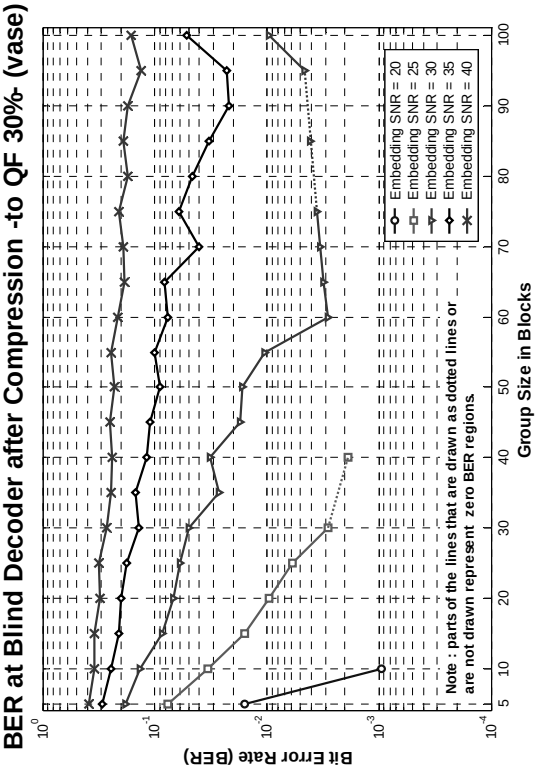


Fig. 4.10 (k)

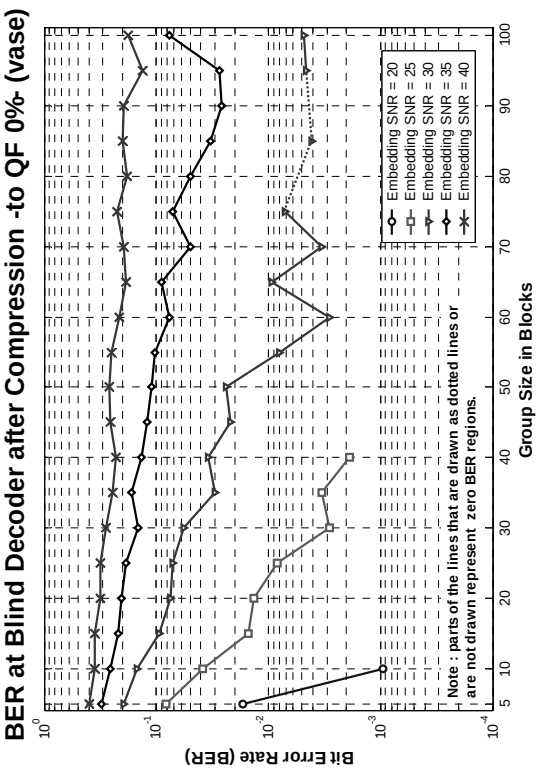


Fig. 4.10 (l)

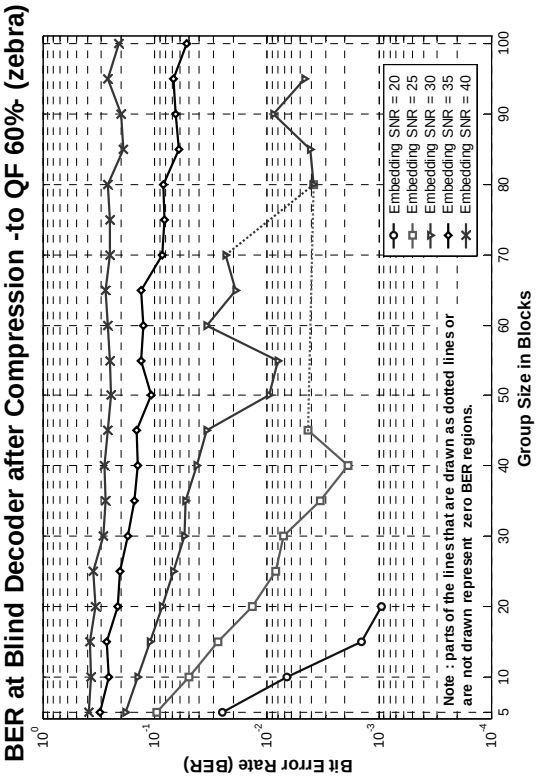


Fig. 4.10 (m)

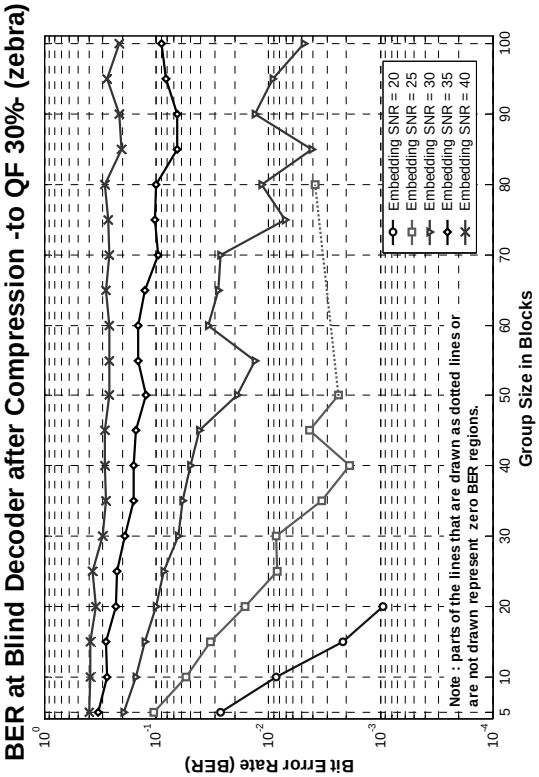


Fig. 4.10 (n)

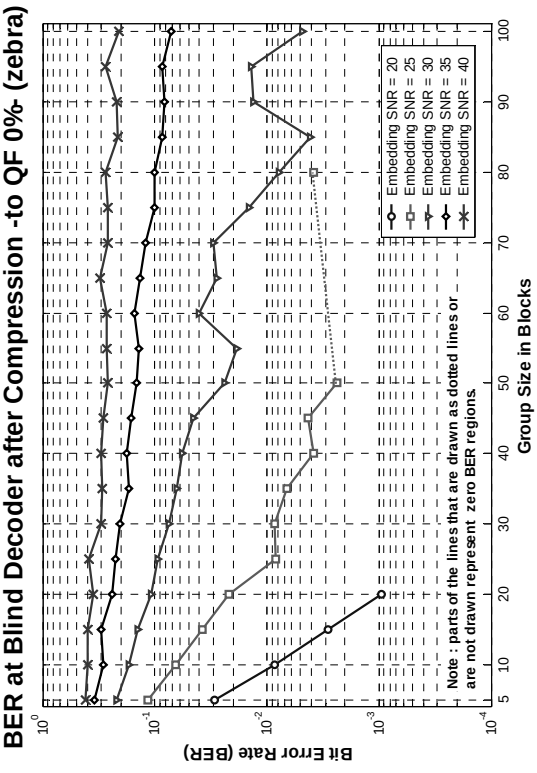


Fig. 4.10 (o)

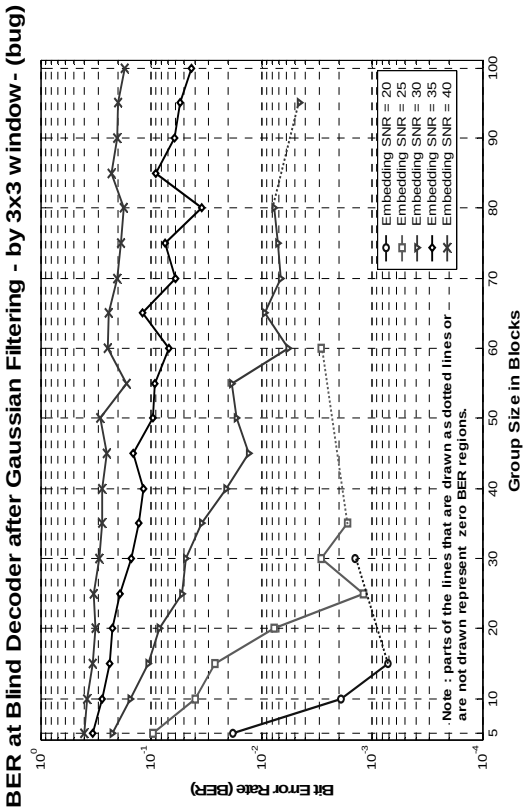


Fig. 4.11 (a)

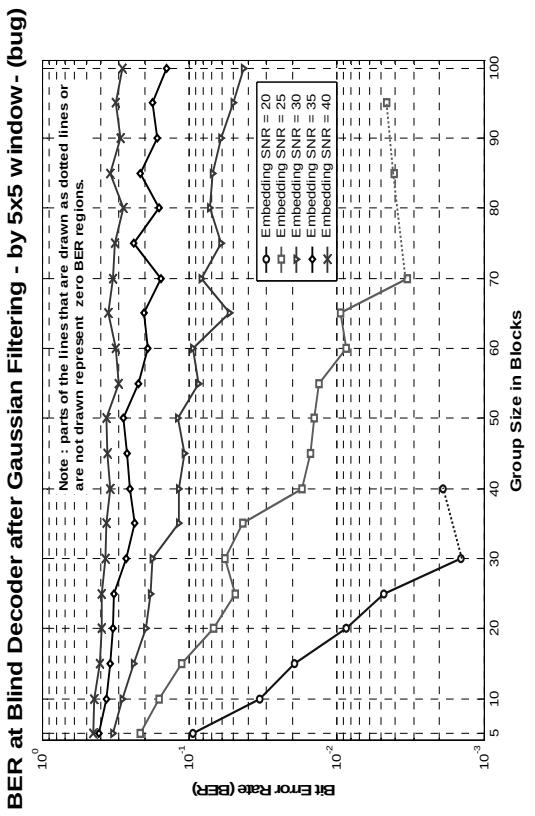


Fig. 4.11 (b)

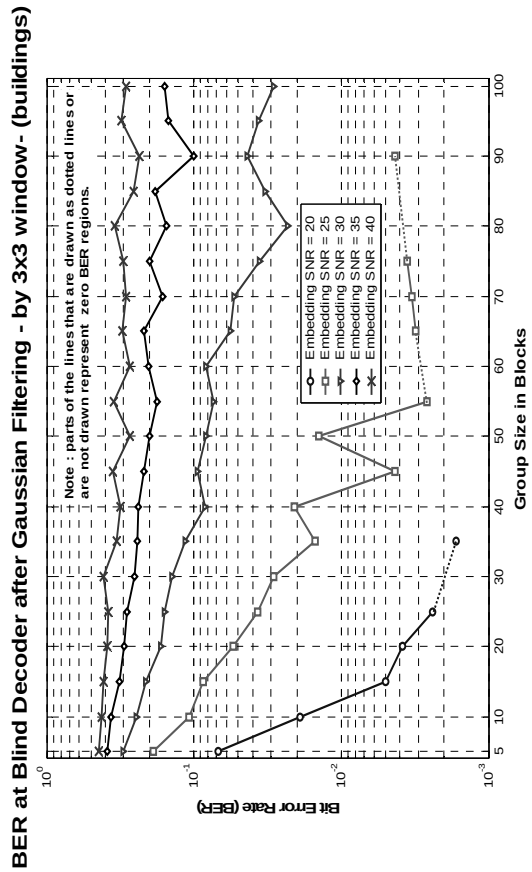


Fig. 4.11 (c)

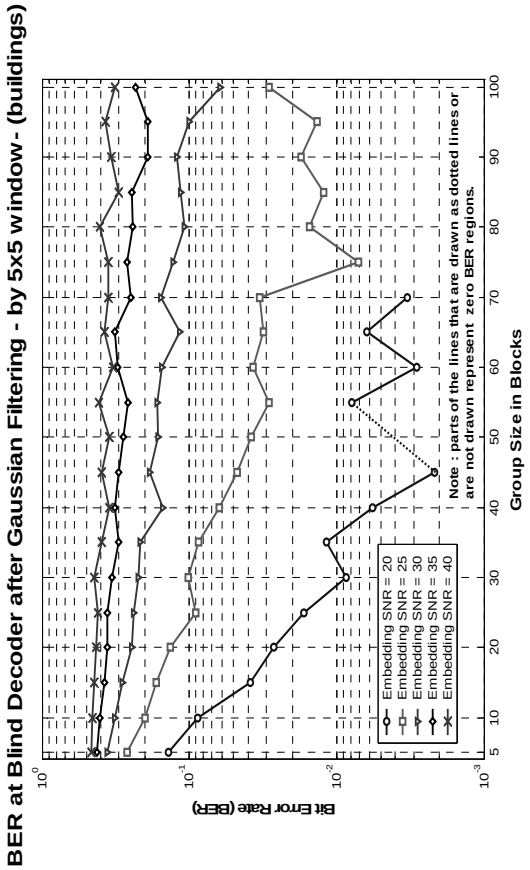


Fig. 4.11 (d)

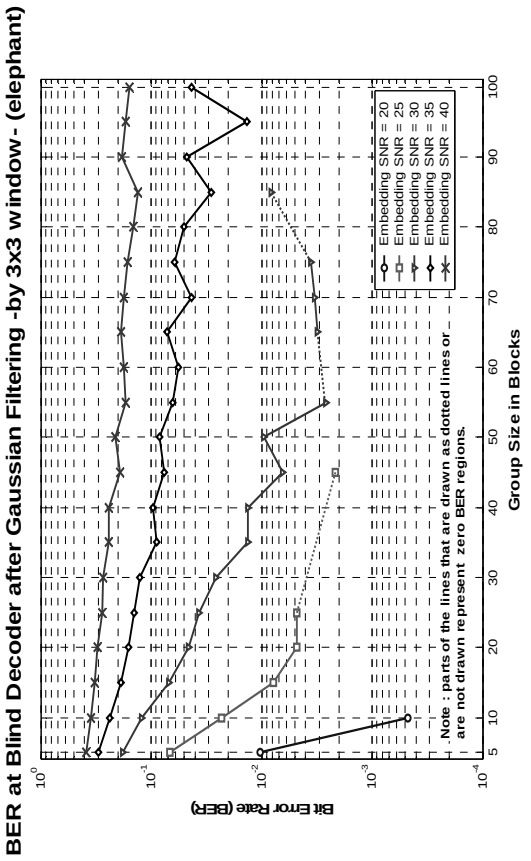


Fig. 4.11 (e)

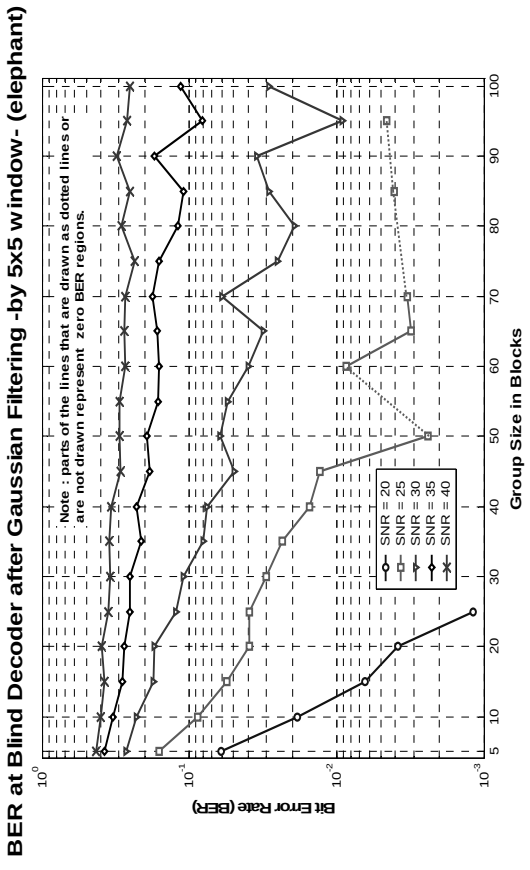


Fig. 4.11 (f)

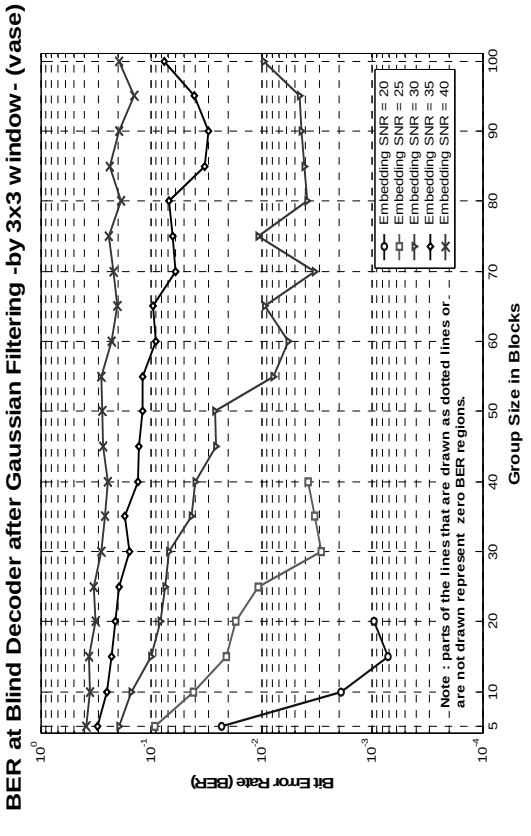


Fig. 4.11 (g)

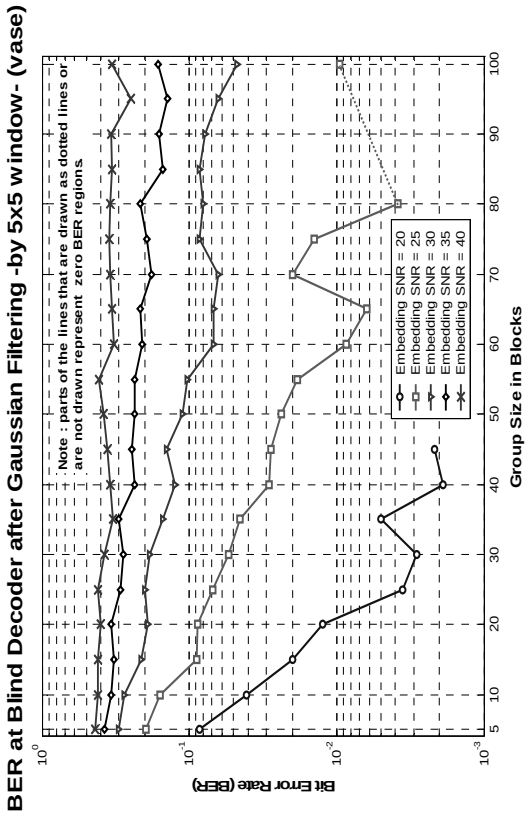


Fig. 4.11 (h)

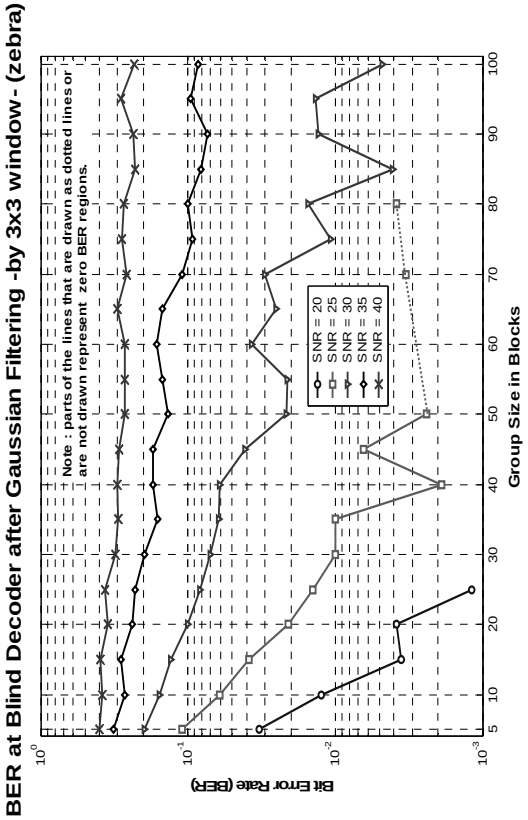


Fig. 4.11 (i)

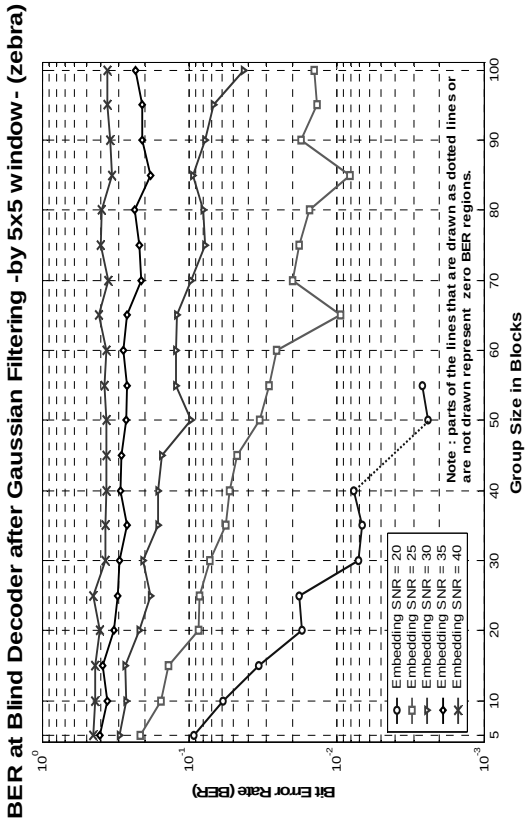


Fig. 4.11 (j)

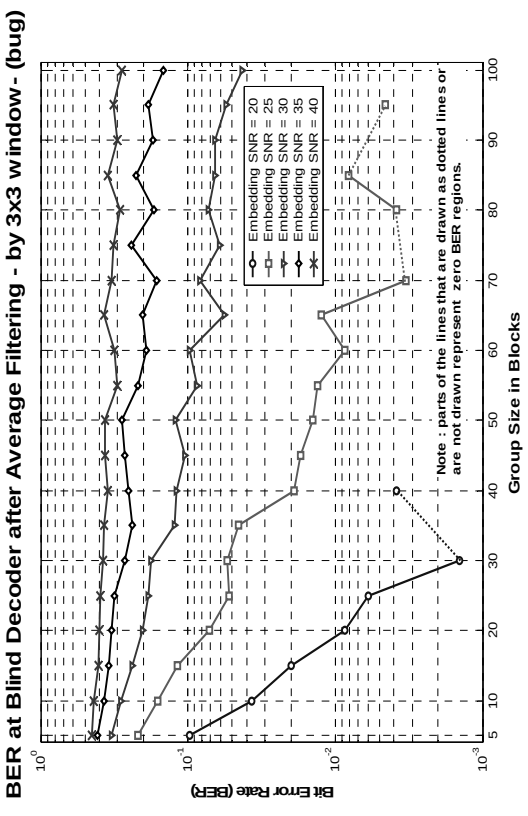


Fig. 4.11 (k)

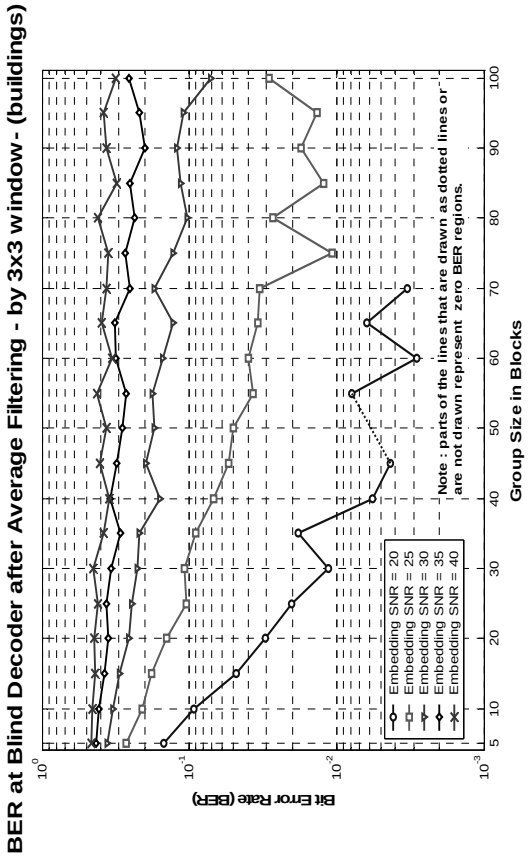


Fig. 4.11 (l)

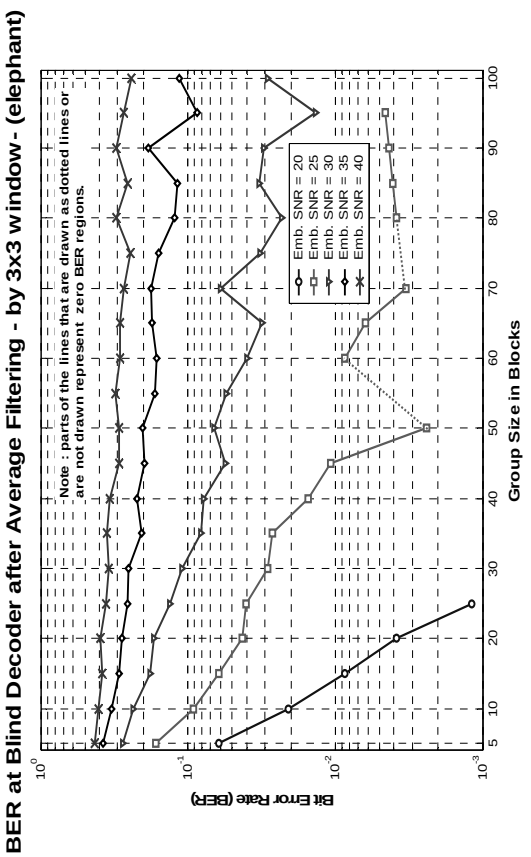


Fig. 4.11 (m)

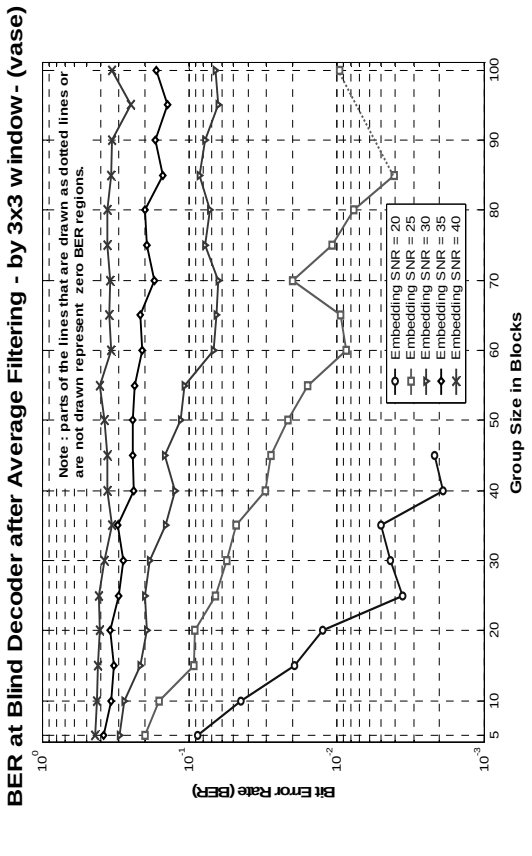


Fig. 4.11 (n)

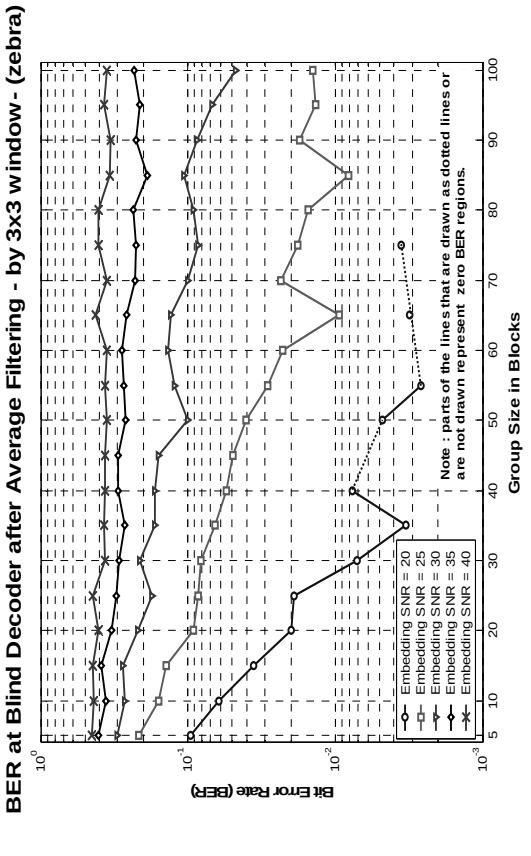


Fig. 4.11 (o)

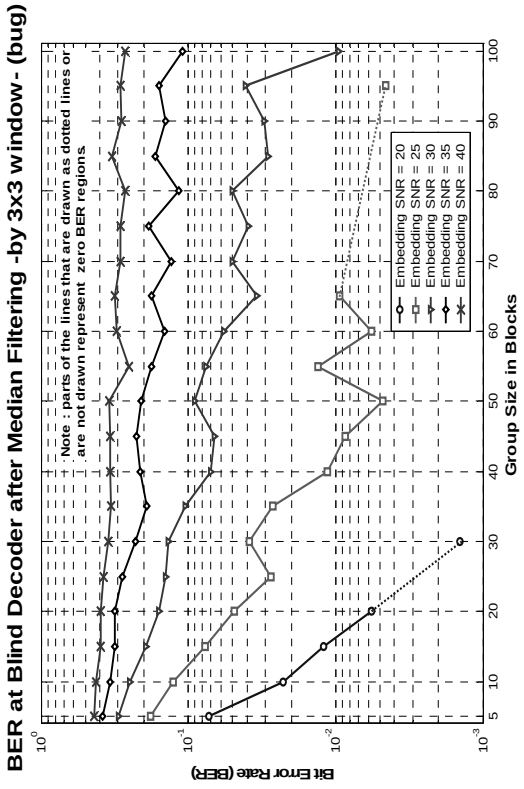


Fig. 4.11 (p)

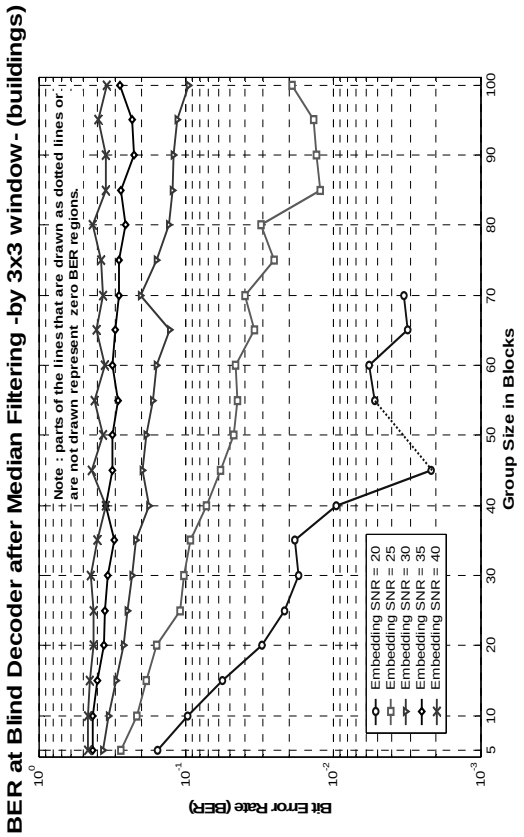


Fig. 4.11 (q)

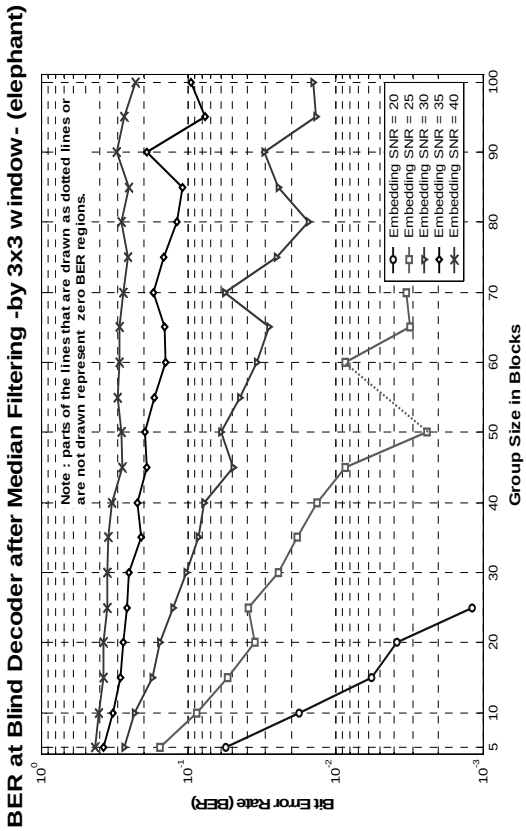


Fig. 4.11 (r)

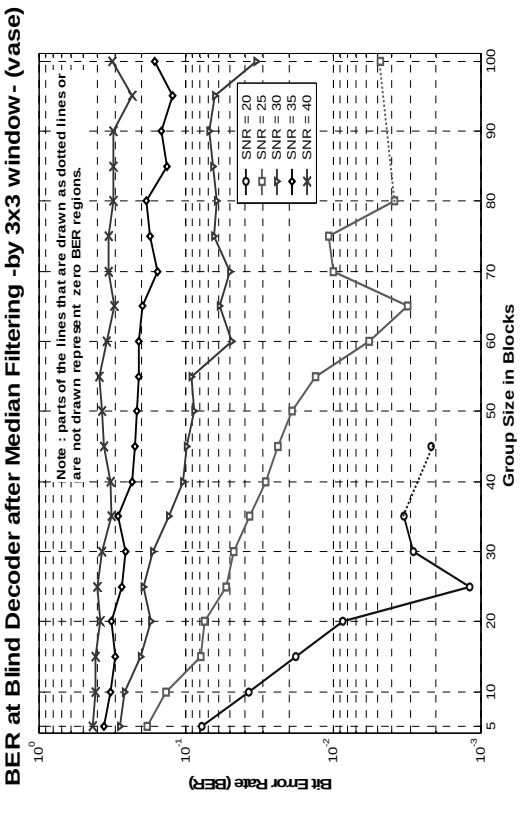


Fig. 4.11 (s)

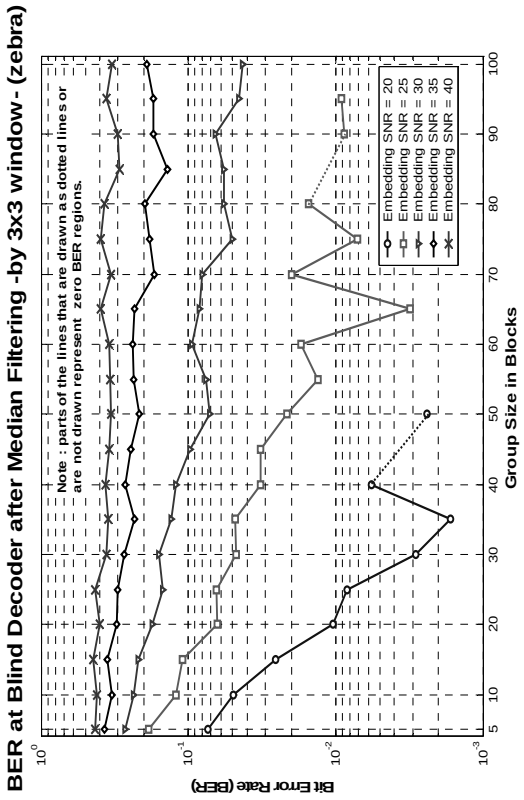


Fig. 4.11 (t)

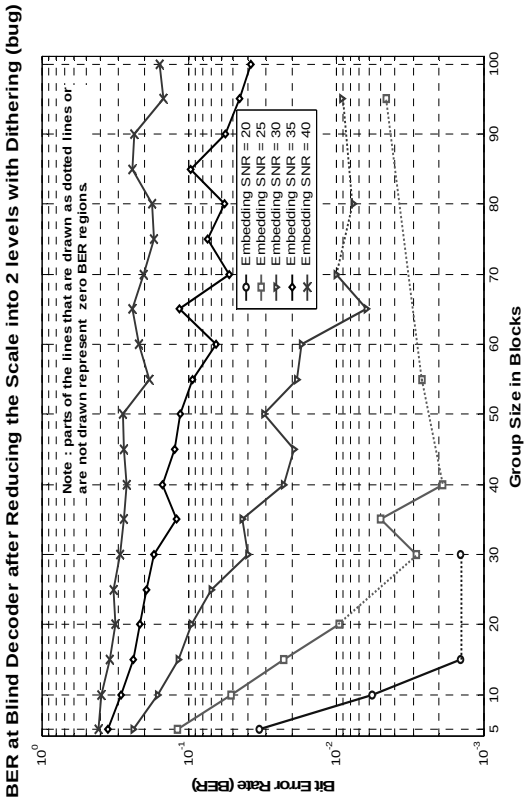


Fig. 4.12 (a)

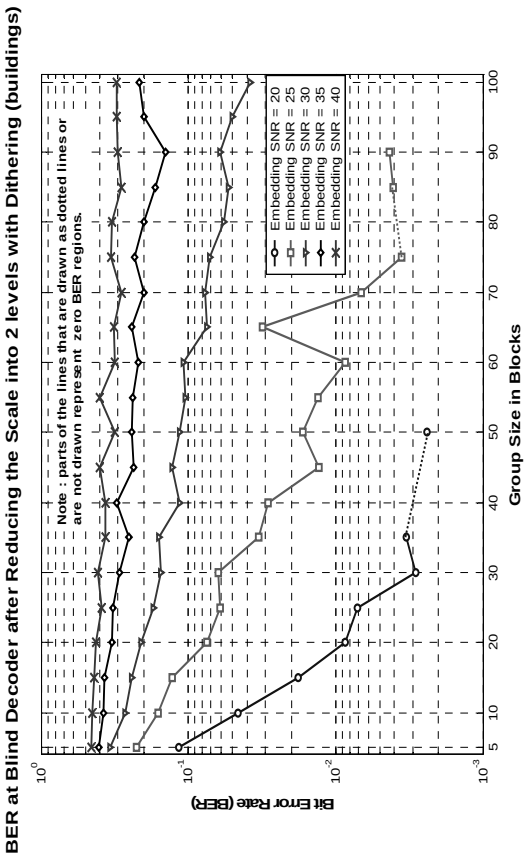


Fig. 4.12 (b)

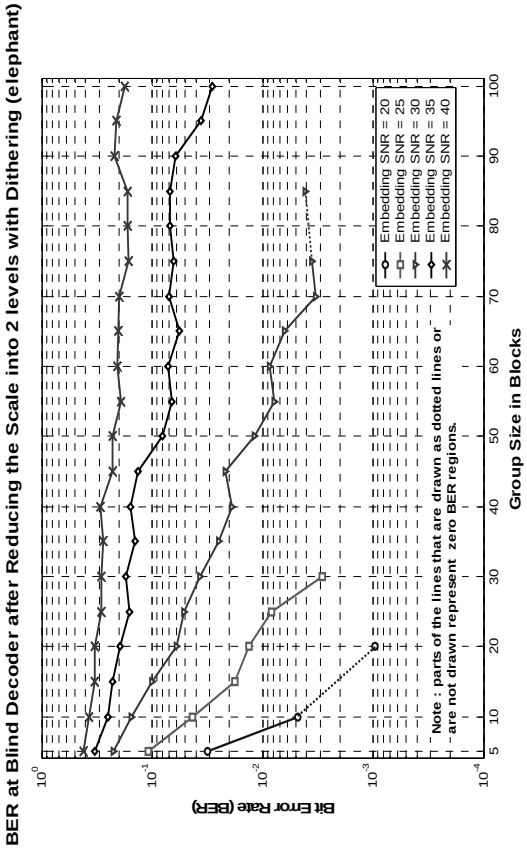


Fig. 4.12 (c)

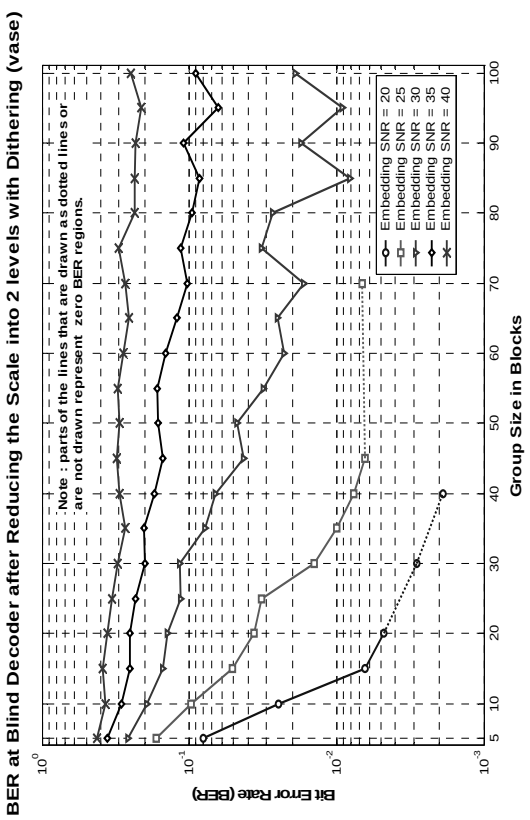


Fig. 4.12 (d)

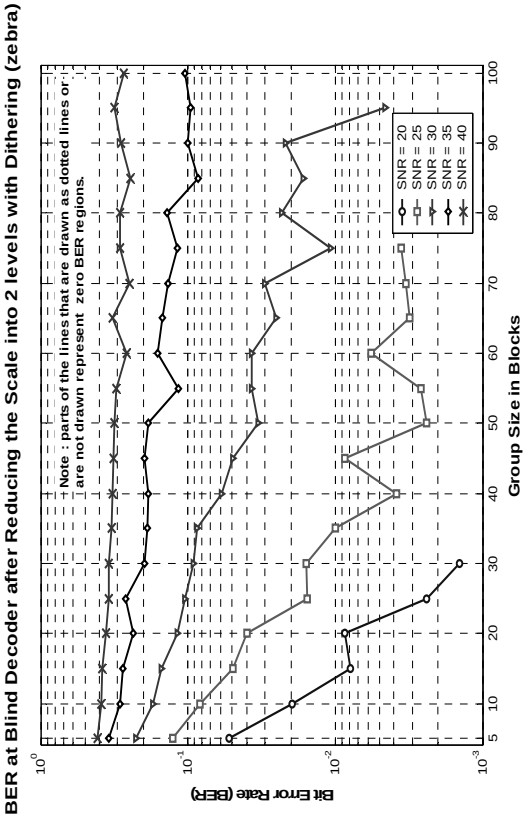


Fig. 4.12 (e)

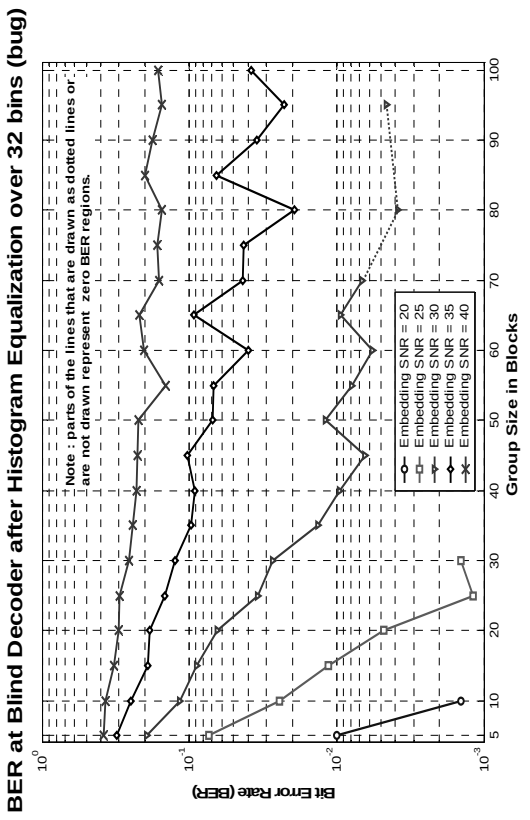


Fig. 4.13 (a)

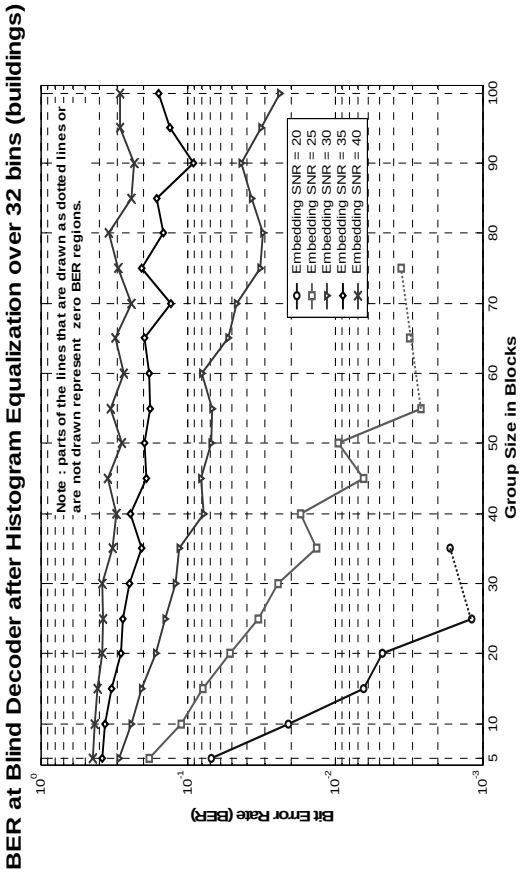


Fig. 4.13 (b)

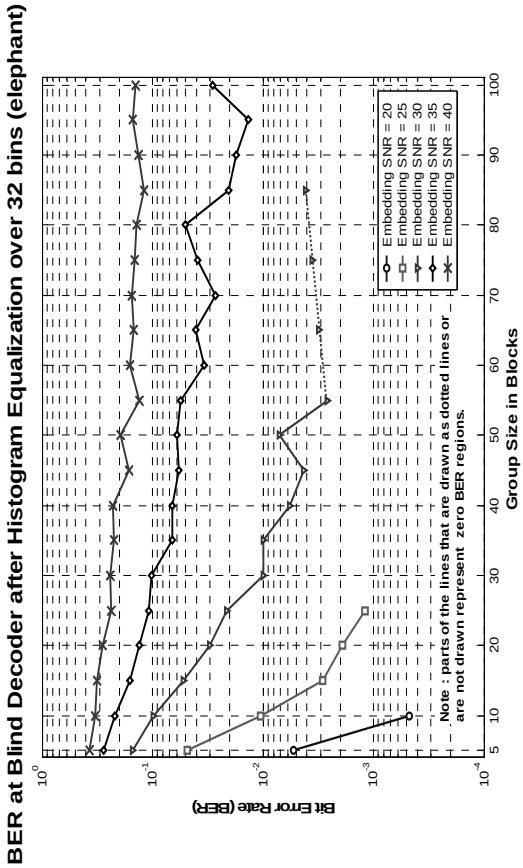


Fig. 4.13 (c)

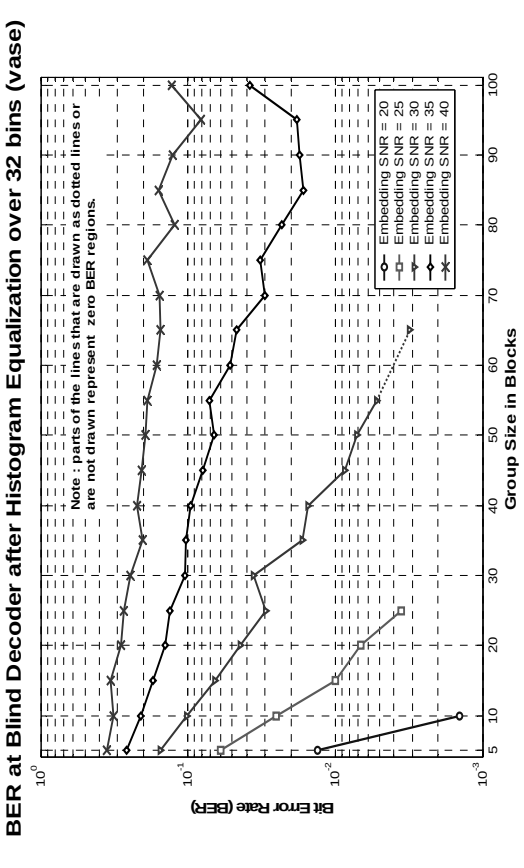


Fig. 4.13 (d)

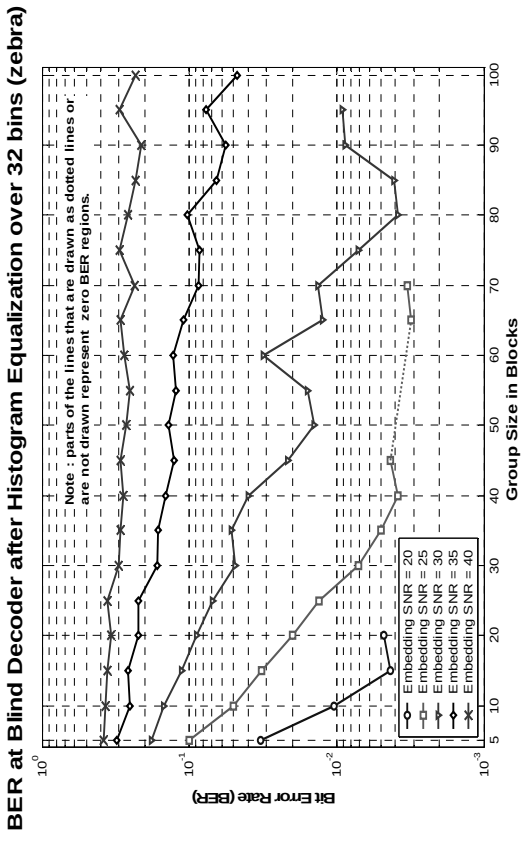


Fig. 4.13 (e)

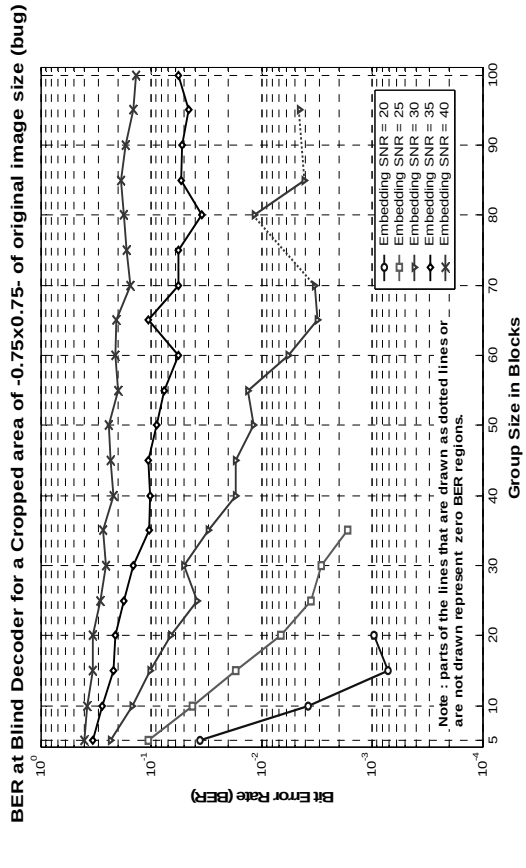


Fig. 4.14 (a)

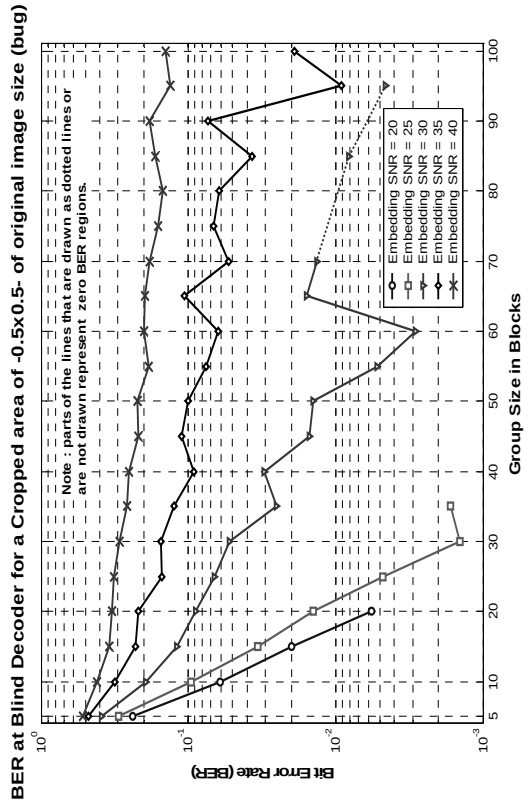


Fig. 4.14 (b)

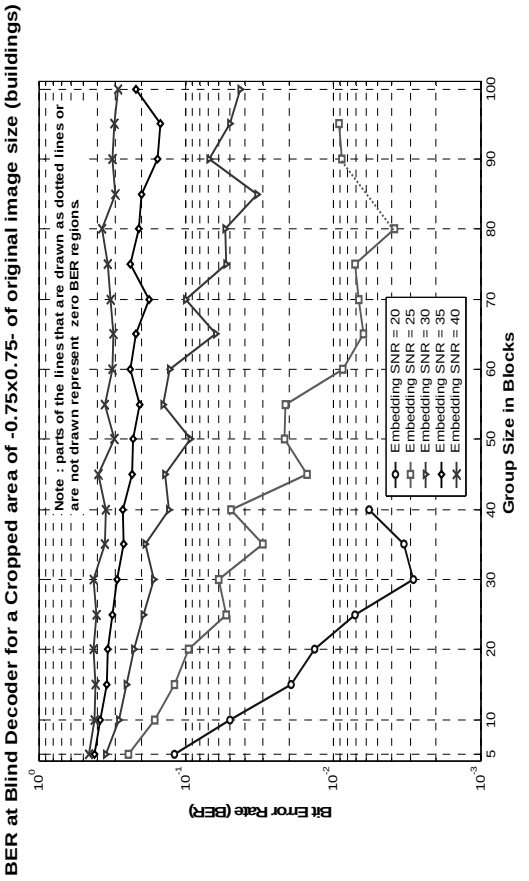


Fig. 4.14 (c)

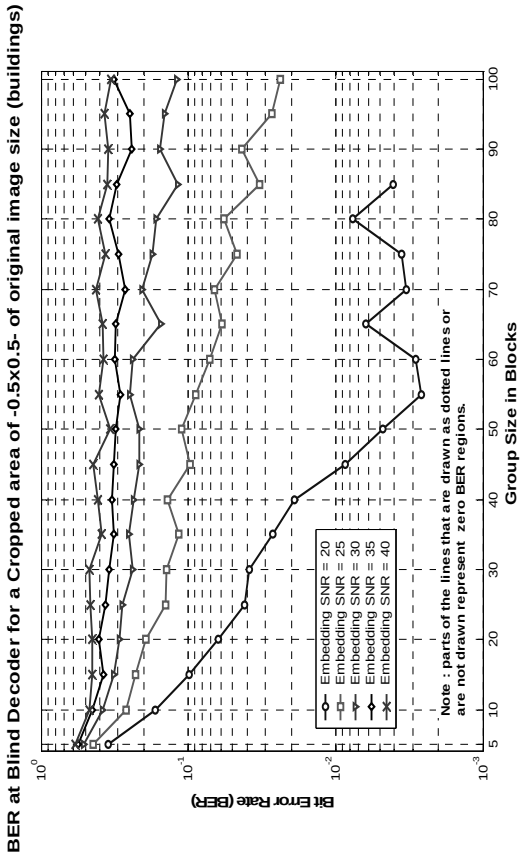


Fig. 4.14 (d)

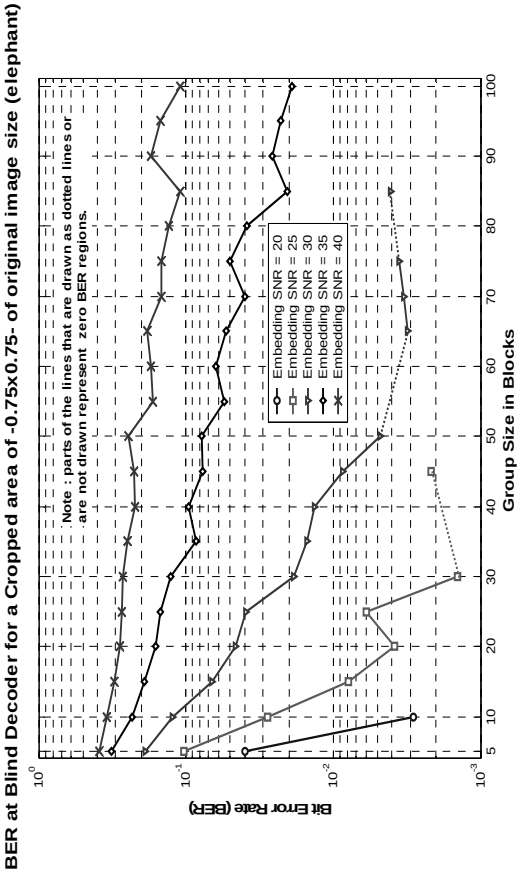


Fig. 4.14 (e)

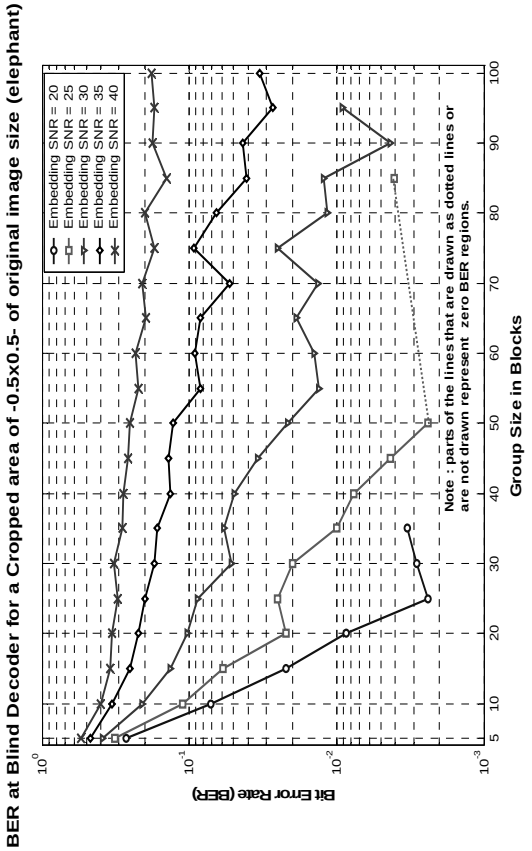


Fig. 4.14 (f)

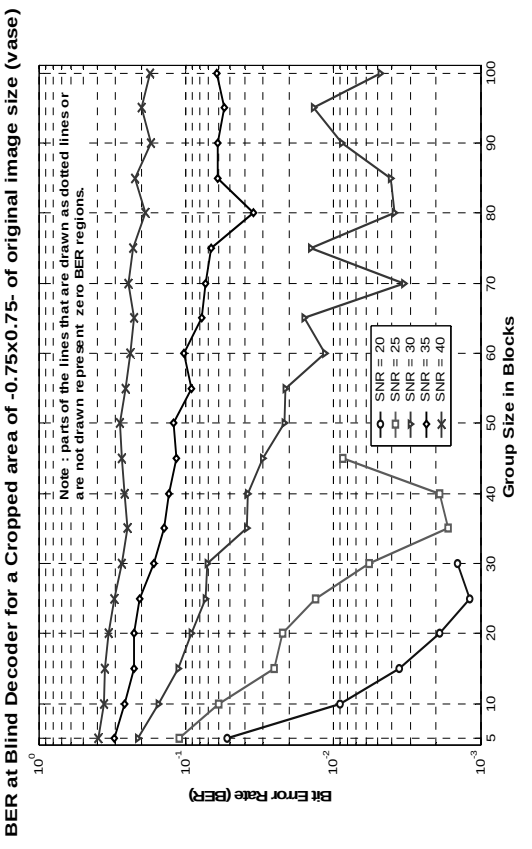


Fig. 4.14 (g)

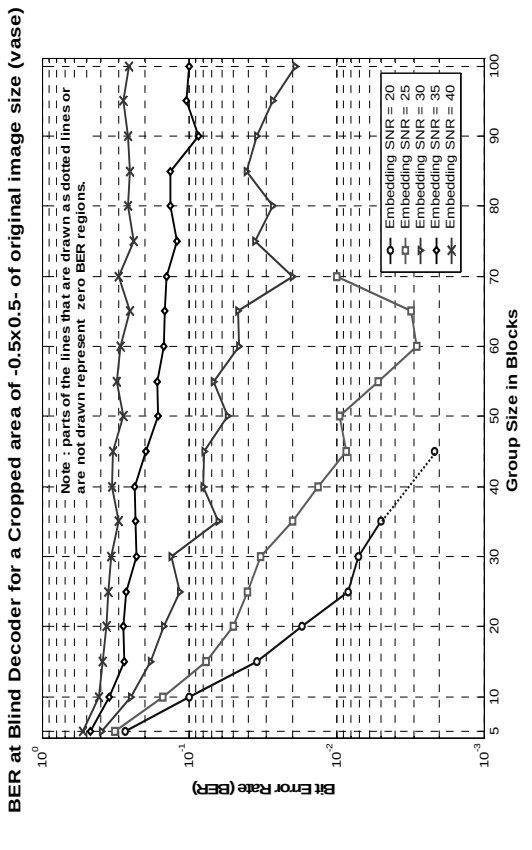


Fig. 4.14 (h)

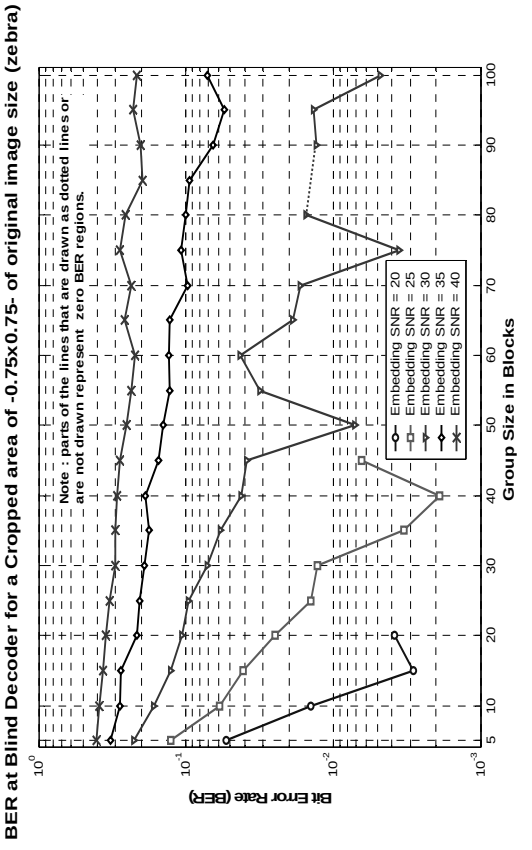


Fig. 4.14 (i)

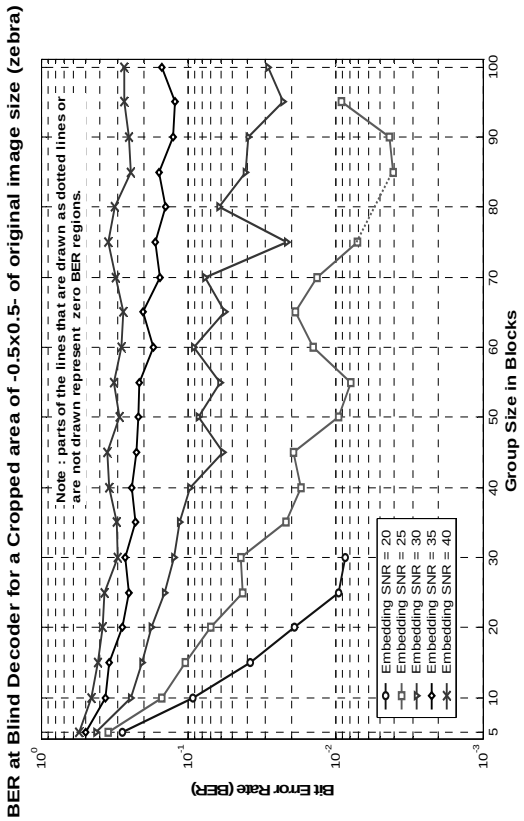


Fig. 4.14 (j)

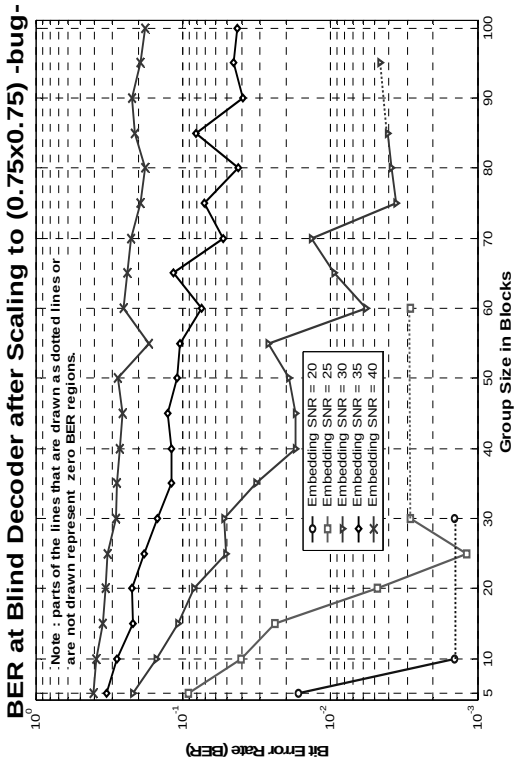


Fig. 4.15 (a)

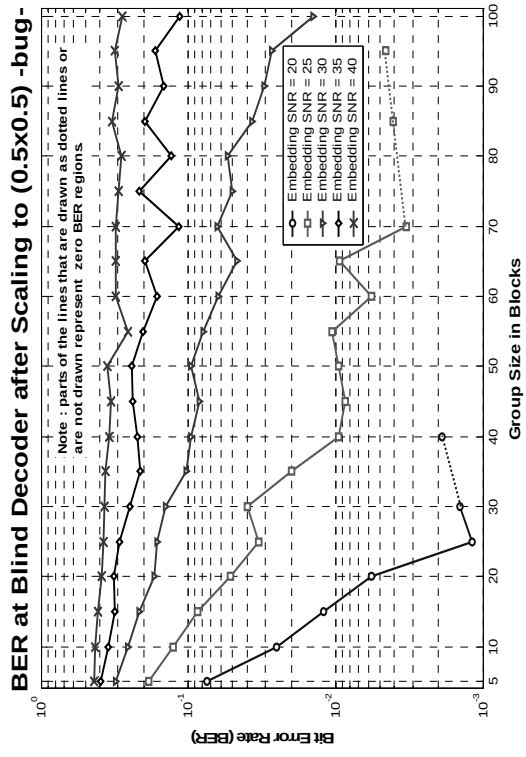


Fig. 4.15 (b)

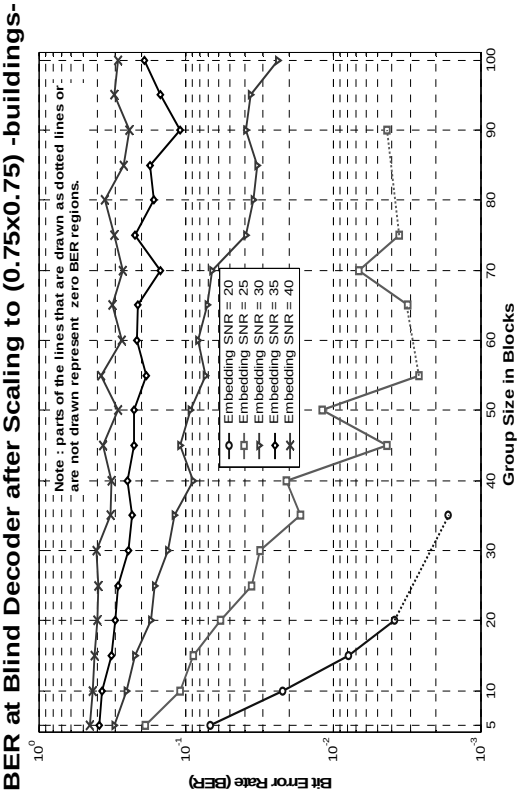


Fig. 4.15 (c)

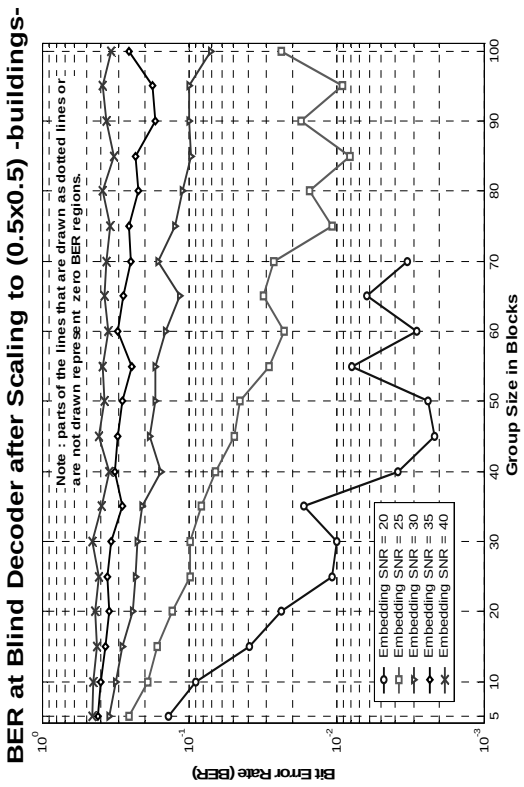


Fig. 4.15 (d)

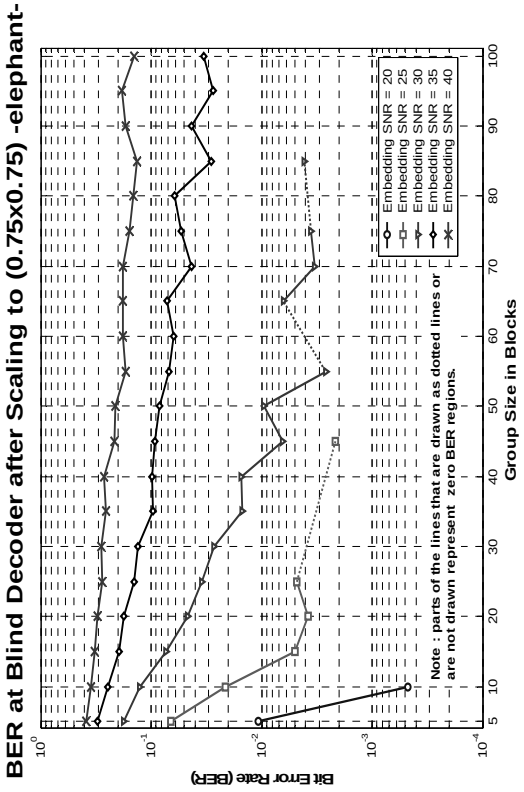


Fig. 4.15 (e)

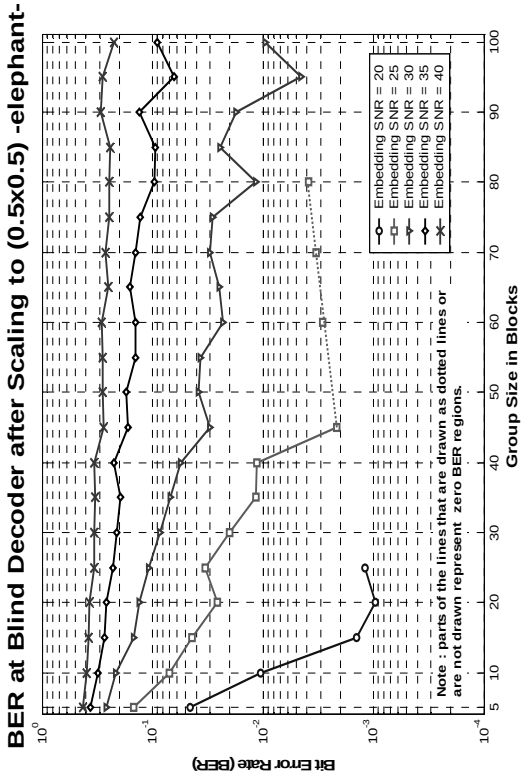


Fig. 4.15 (f)

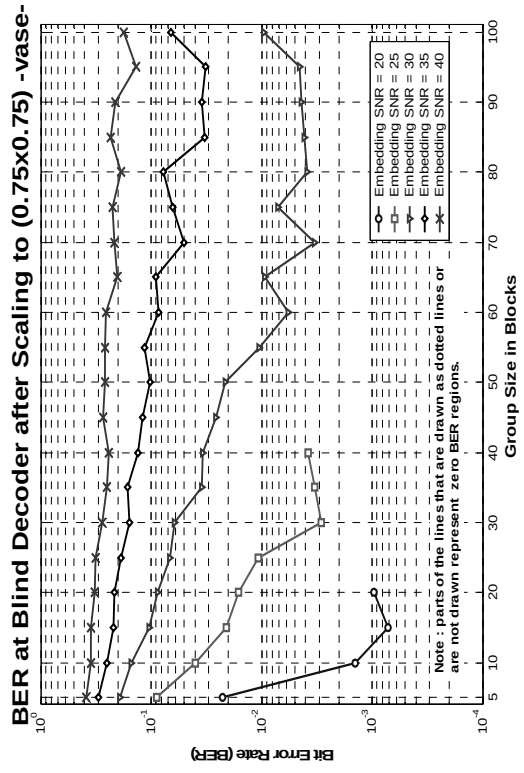


Fig. 4.15 (g)

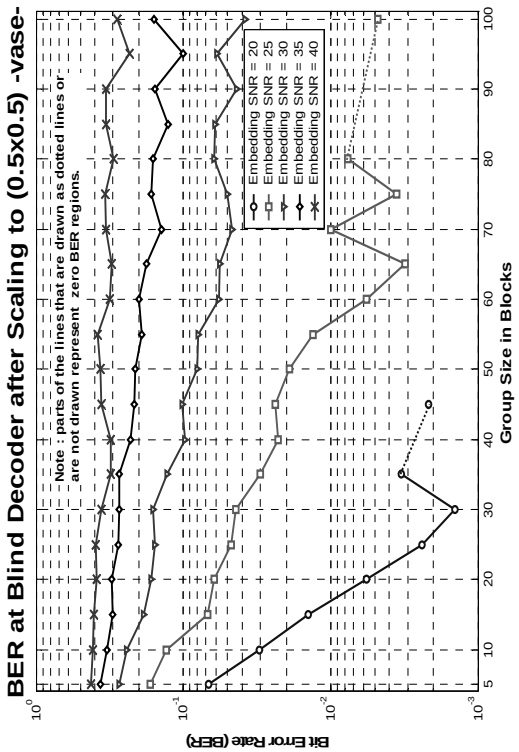


Fig. 4.15 (h)

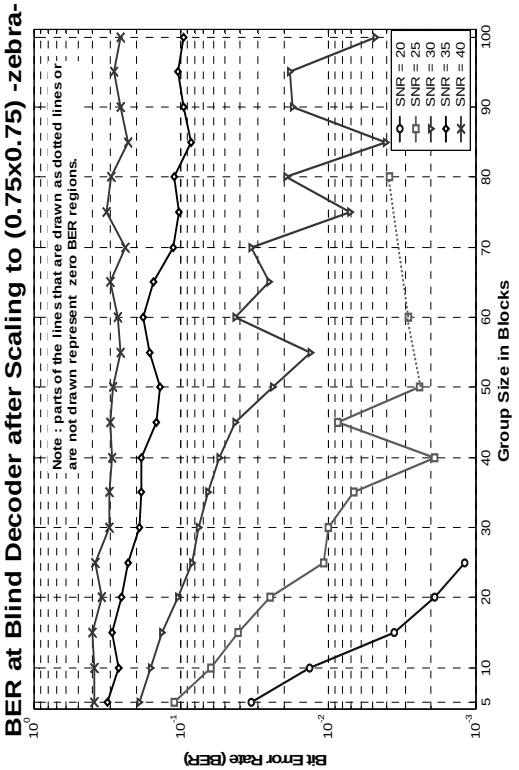


Fig. 4.15 (i)

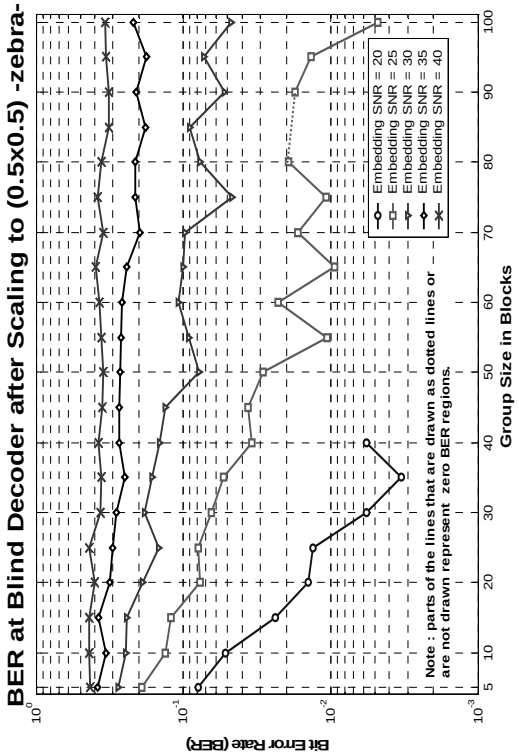


Fig. 4.15 (j)

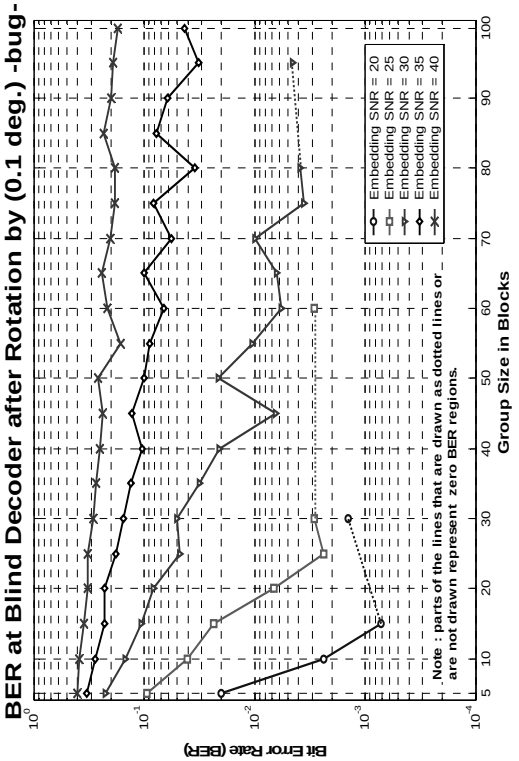


Fig. 4.16 (a)

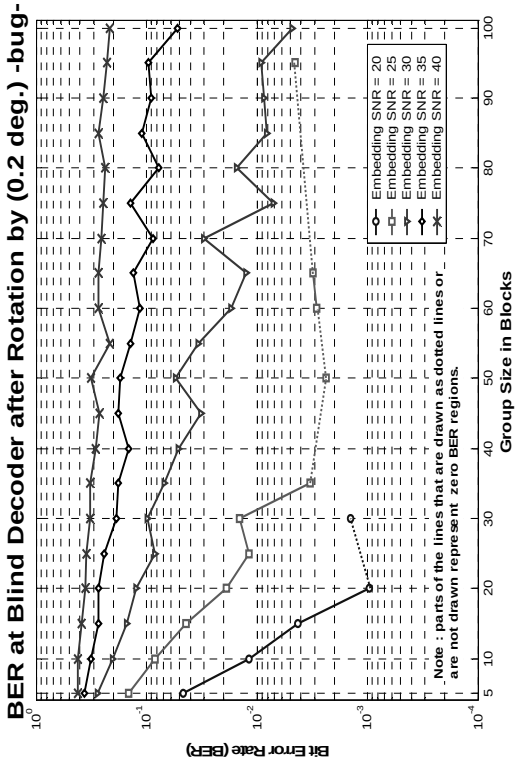


Fig. 4.16 (b)

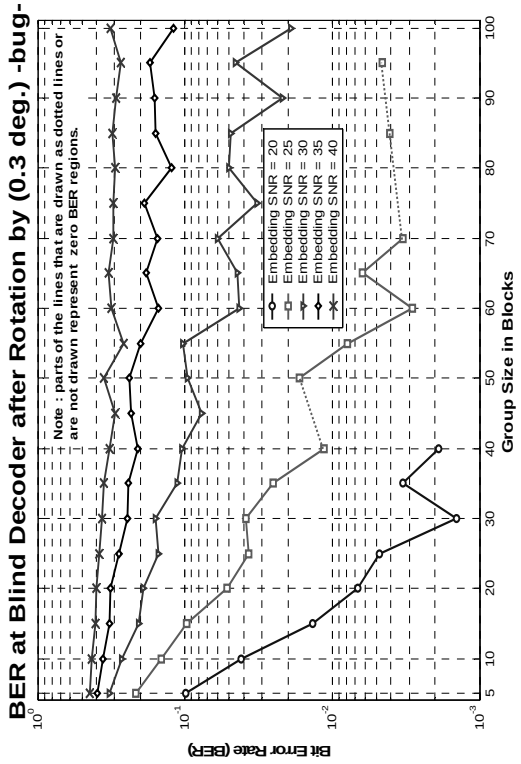


Fig. 4.16 (c)

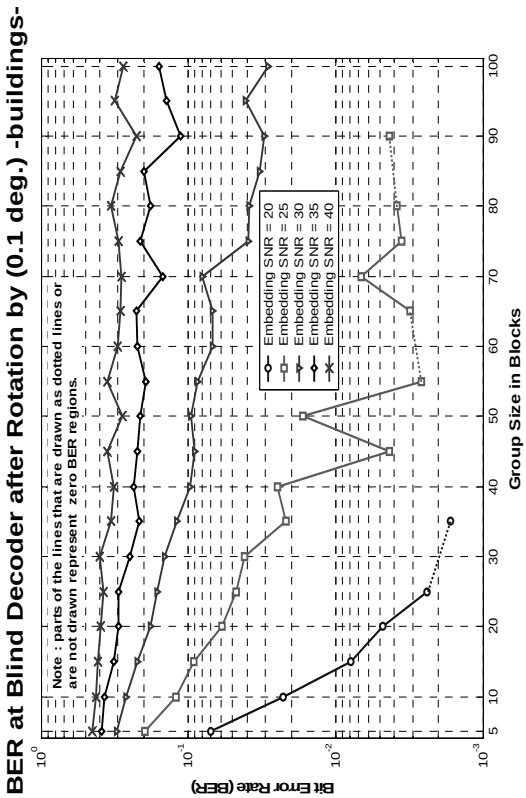


Fig. 4.16 (d)

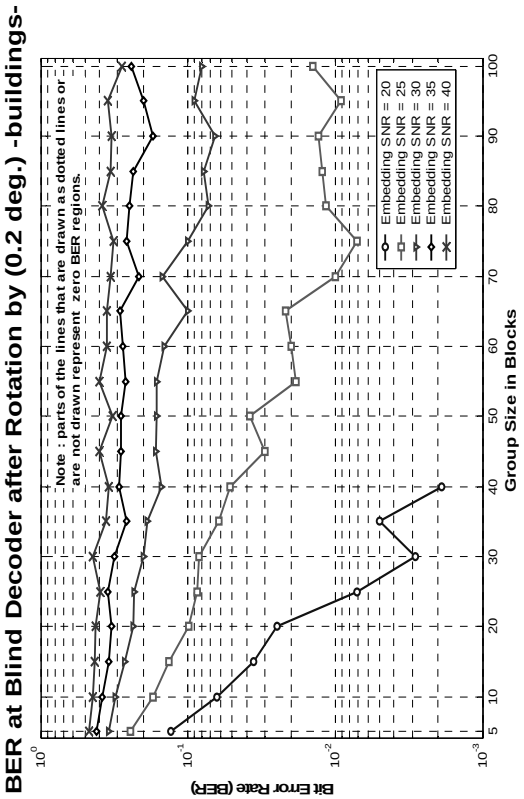


Fig. 4.16 (e)

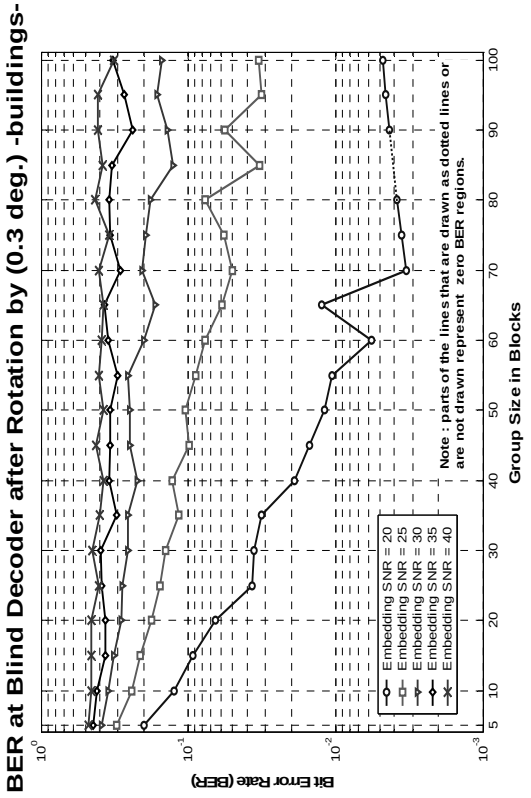


Fig. 4.16 (f)

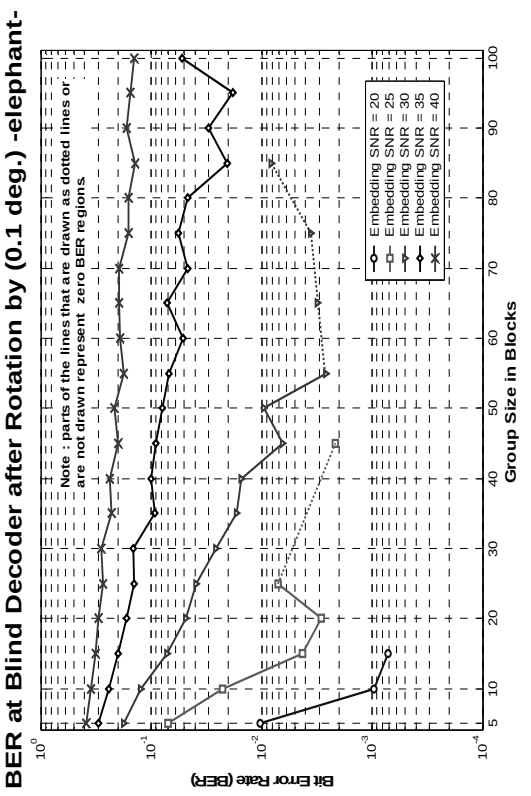


Fig. 4.16 (g)

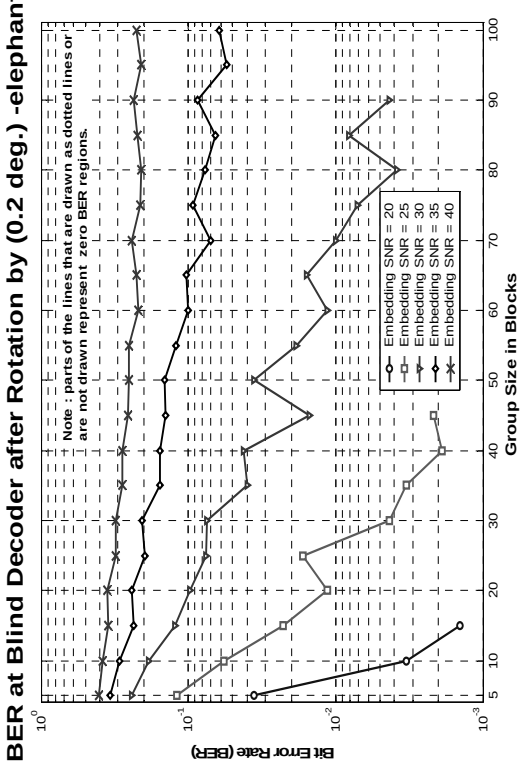


Fig. 4.16 (h)

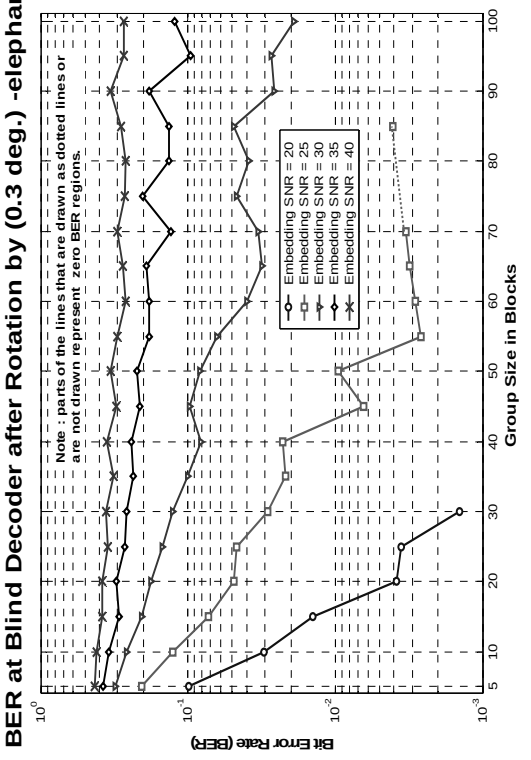


Fig. 4.16 (i)

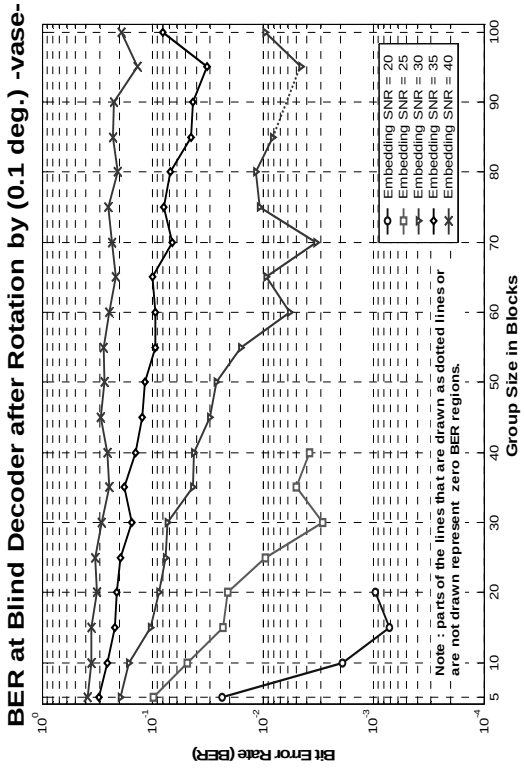


Fig. 4.16 (j)

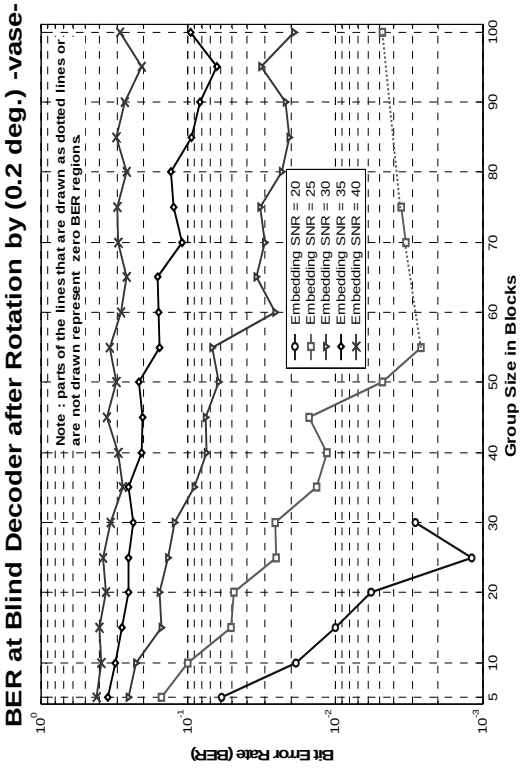


Fig. 4.16 (k)

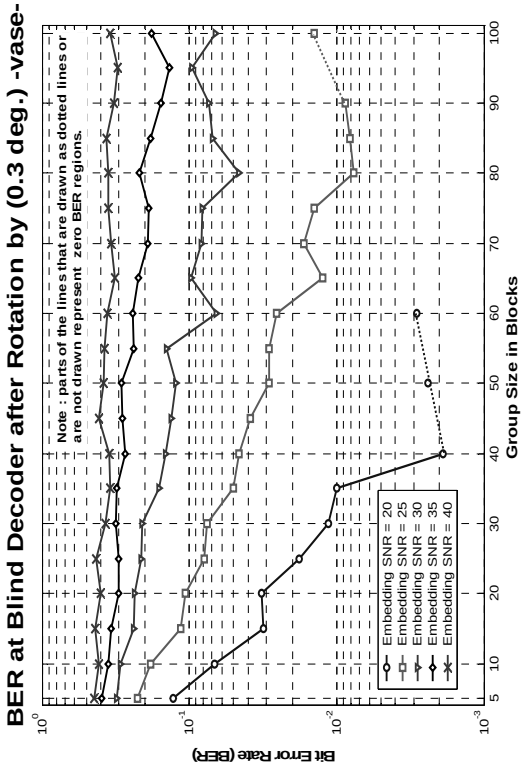


Fig. 4.16 (l)

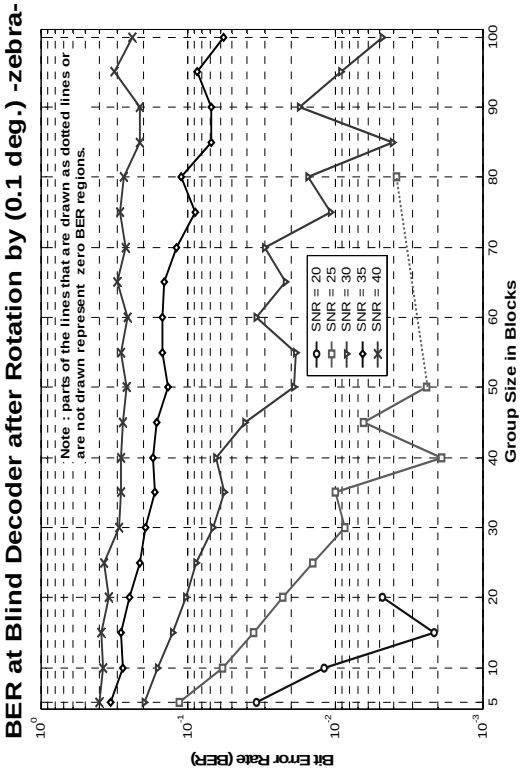


Fig. 4.16 (m)

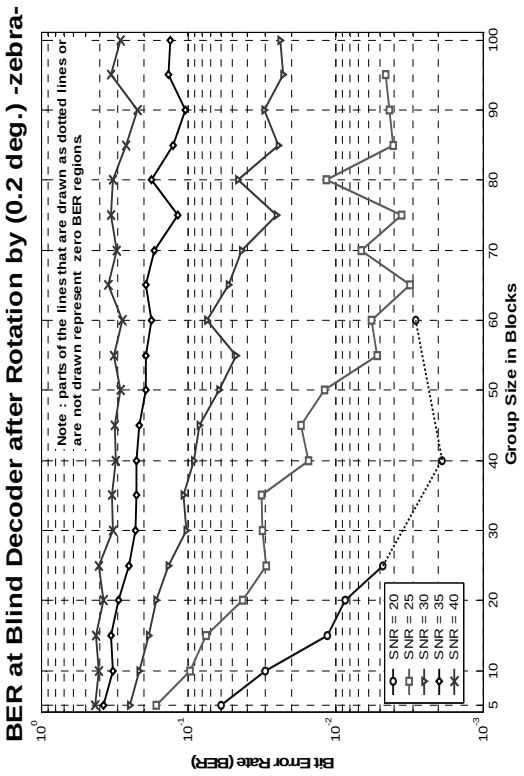


Fig. 4.16 (n)

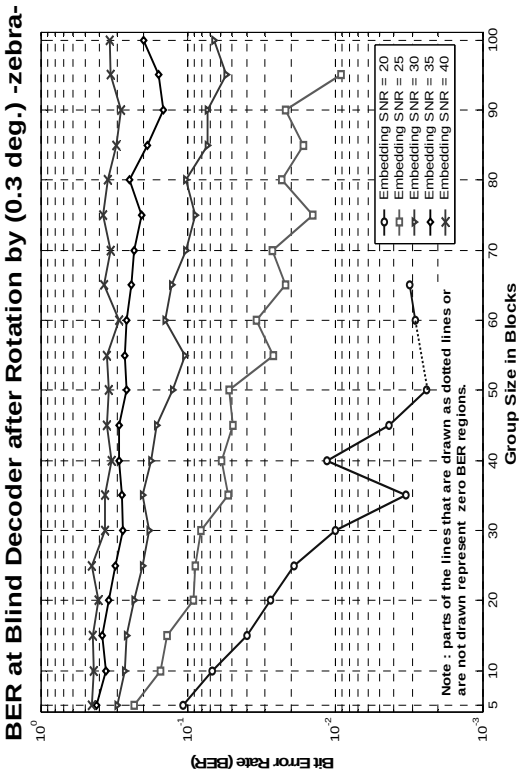


Fig. 4.16 (o)

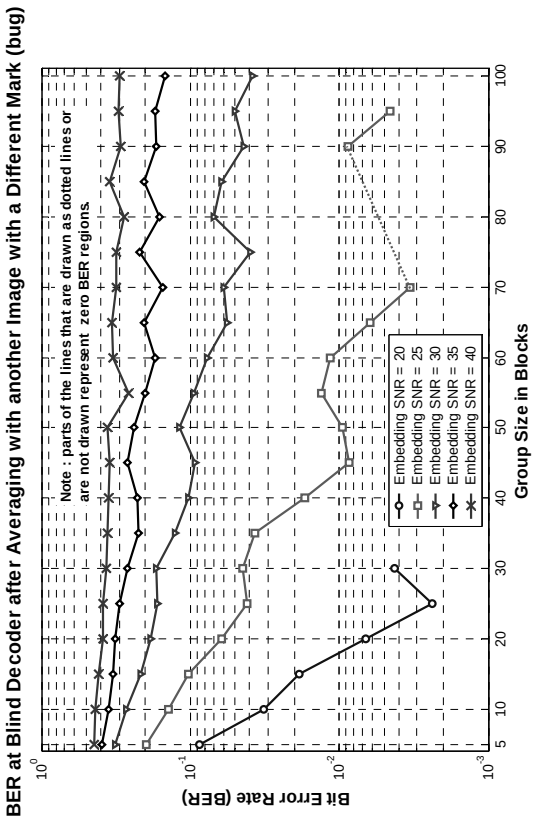


Fig. 4.17 (a)

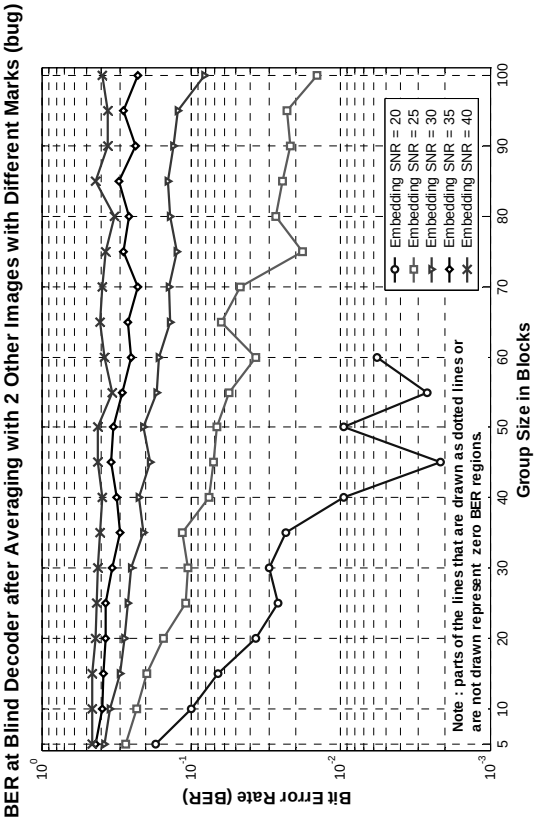


Fig. 4.17 (b)

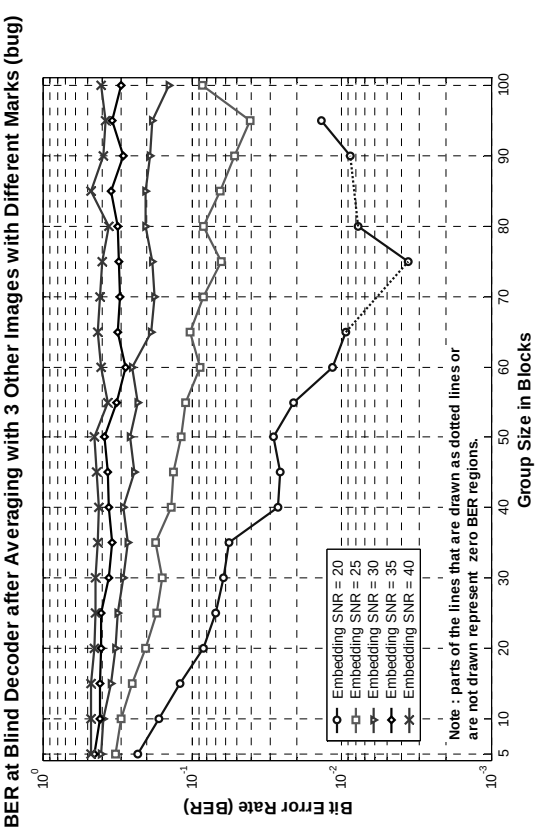


Fig. 4.17 (c)

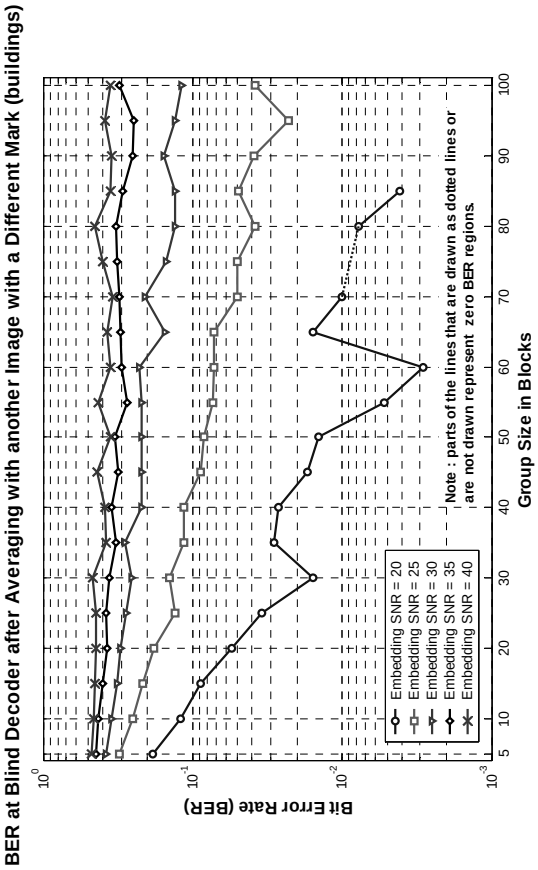


Fig. 4.17 (d)

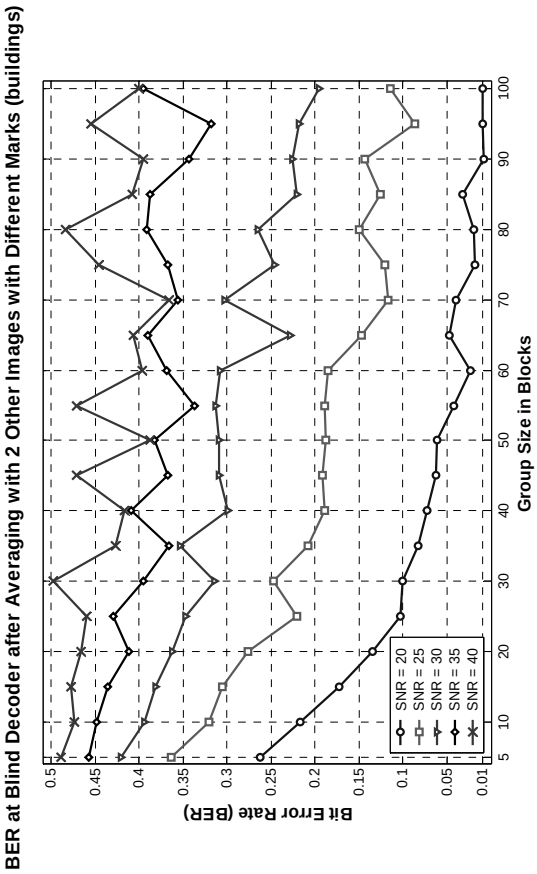


Fig. 4.17 (e)

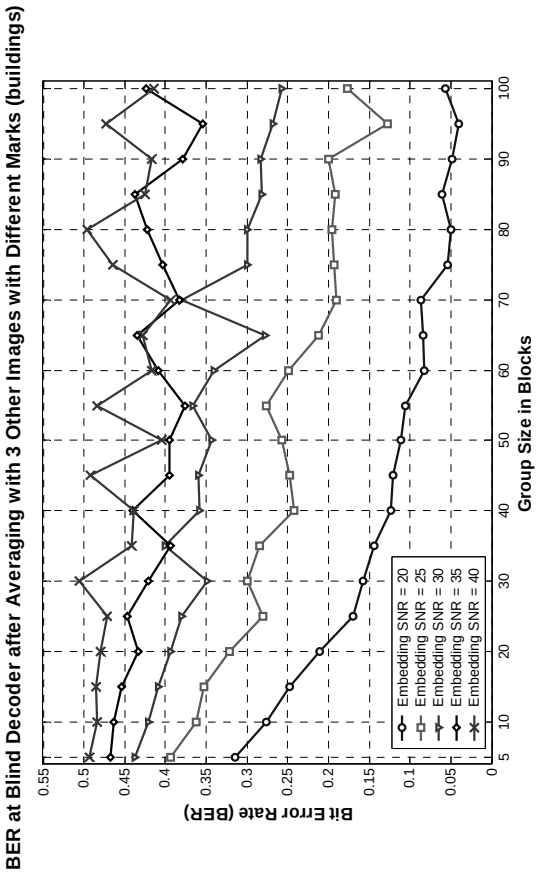


Fig. 4.17 (f)

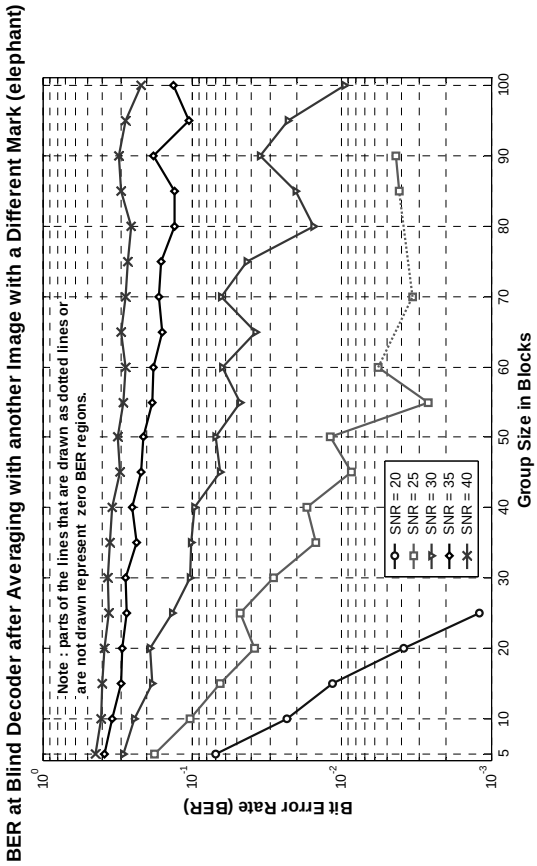


Fig. 4.17 (g)

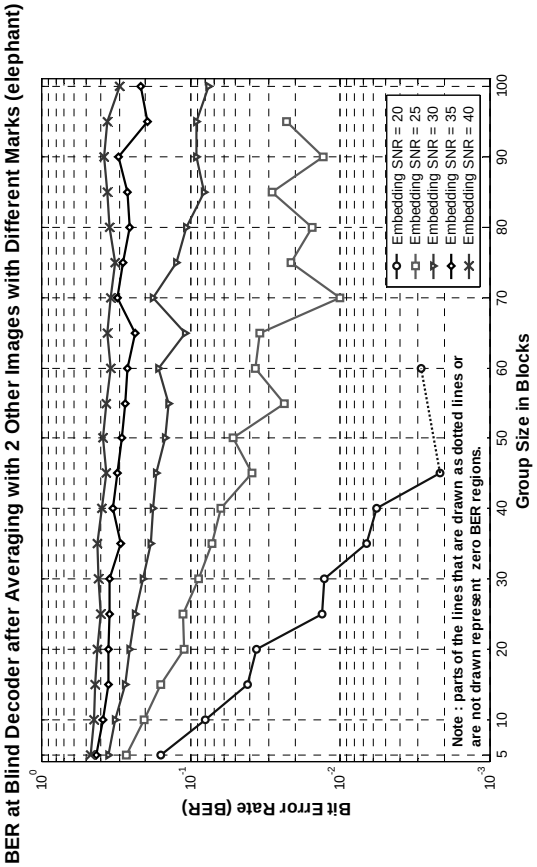


Fig. 4.17 (h)

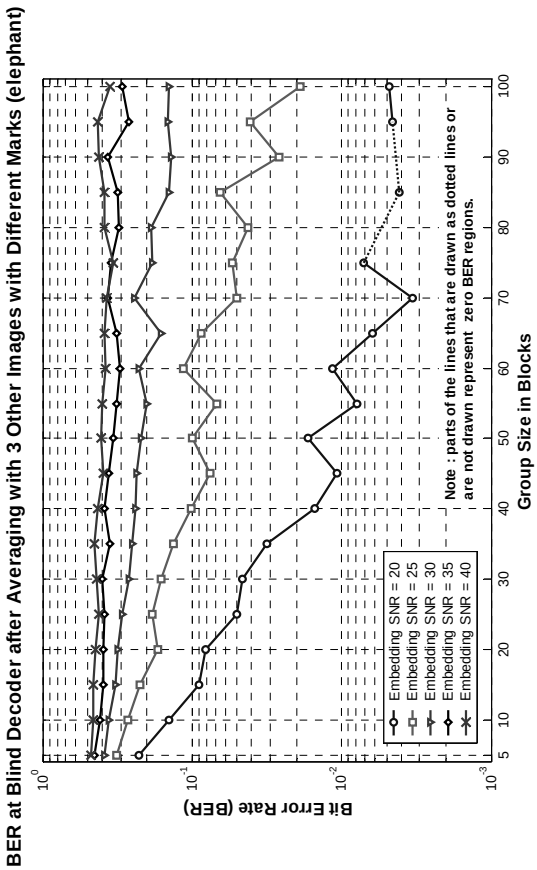


Fig. 4.17 (i)

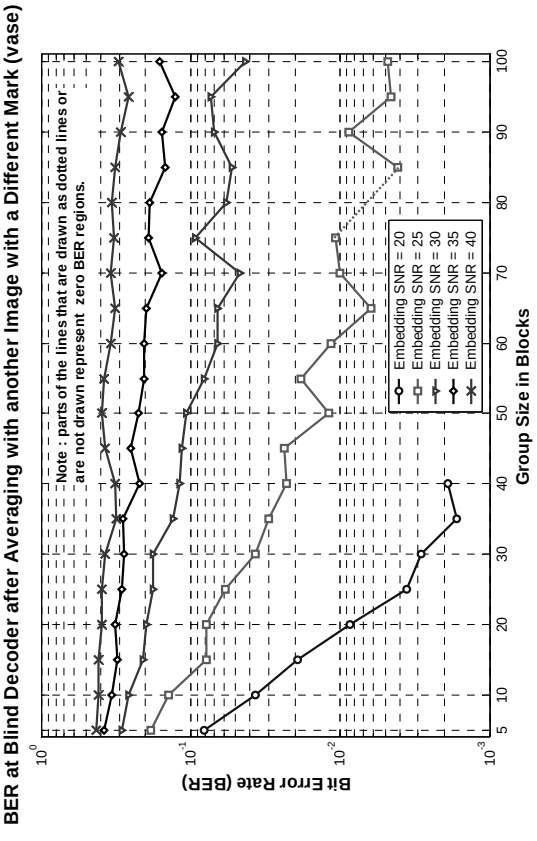


Fig. 4.17 (j)

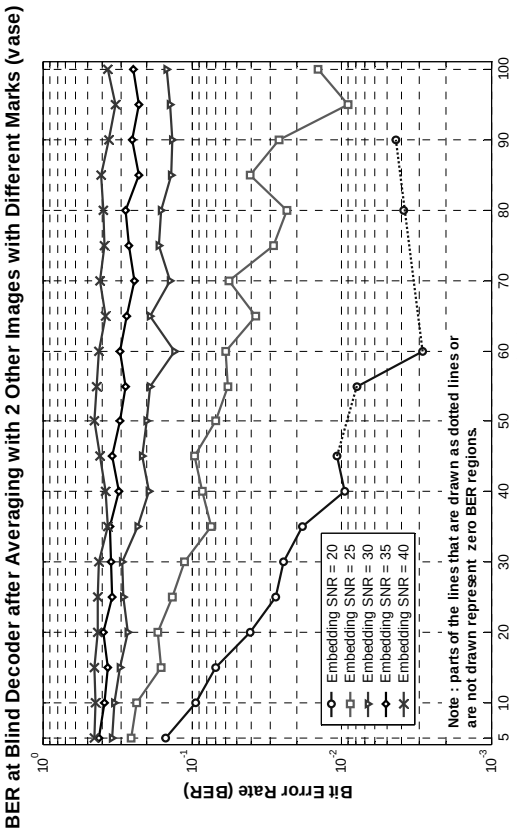


Fig. 4.17 (k)

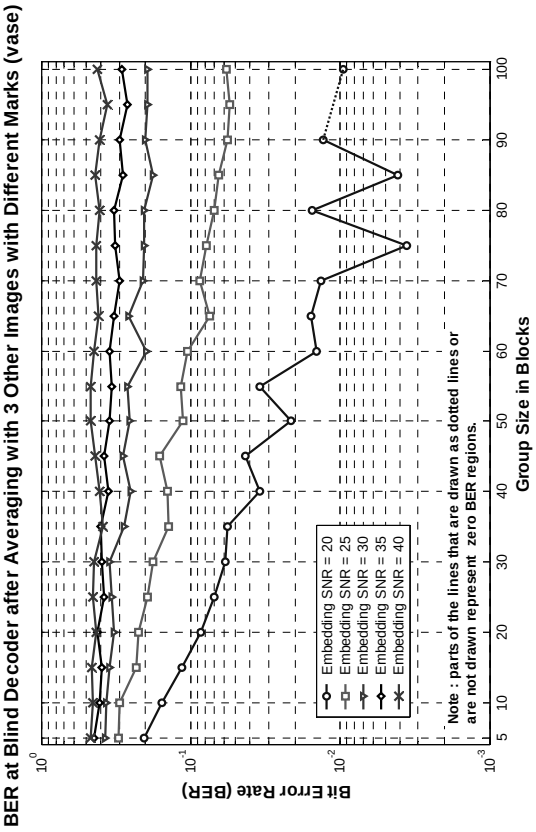


Fig. 4.17 (l)

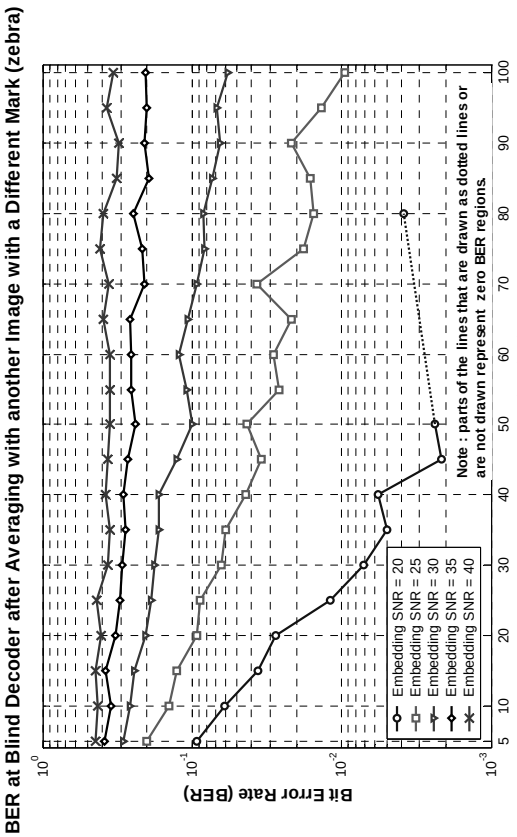


Fig. 4.17 (m)

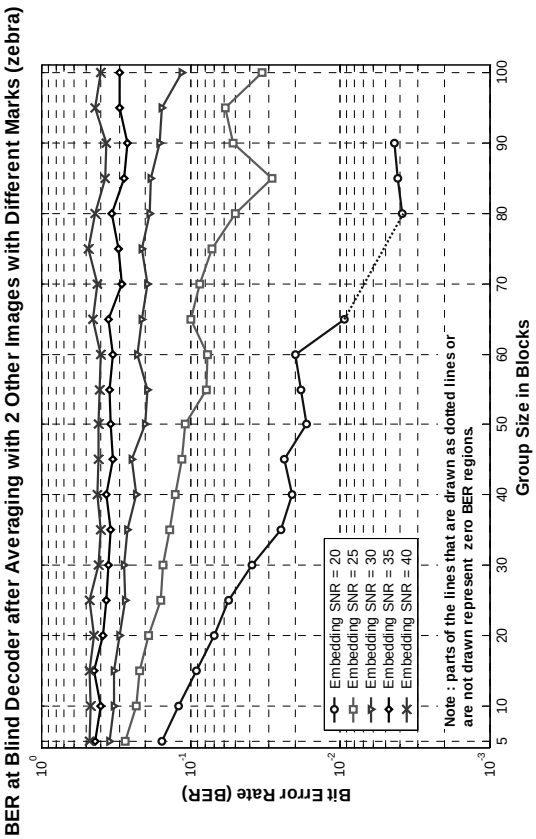


Fig. 4.17 (n)

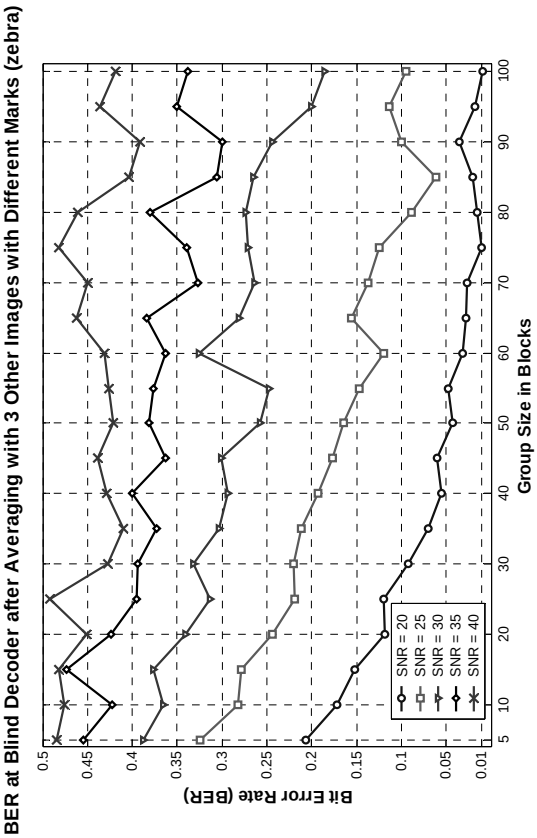


Fig. 4.17 (o)

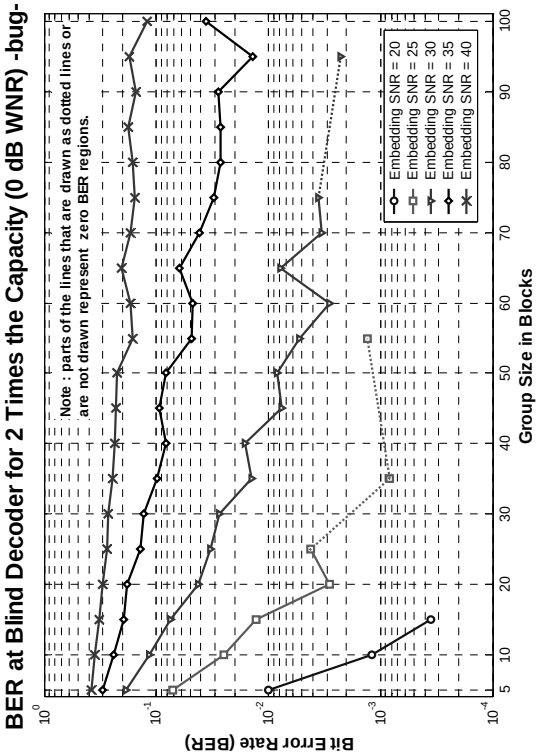


Fig. 4.18 (a)

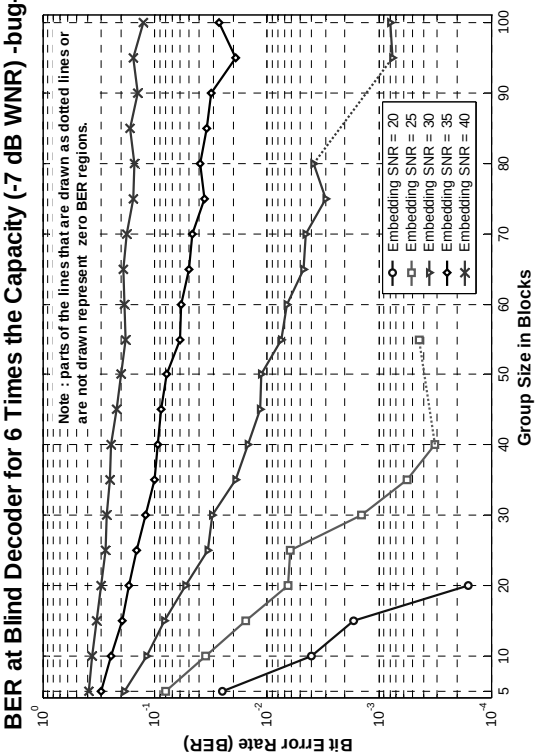


Fig. 4.18 (b)

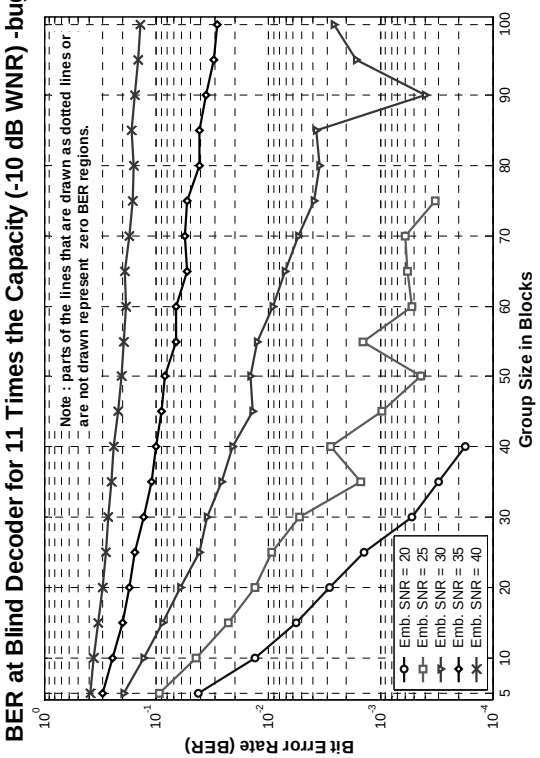


Fig. 4.18 (c)

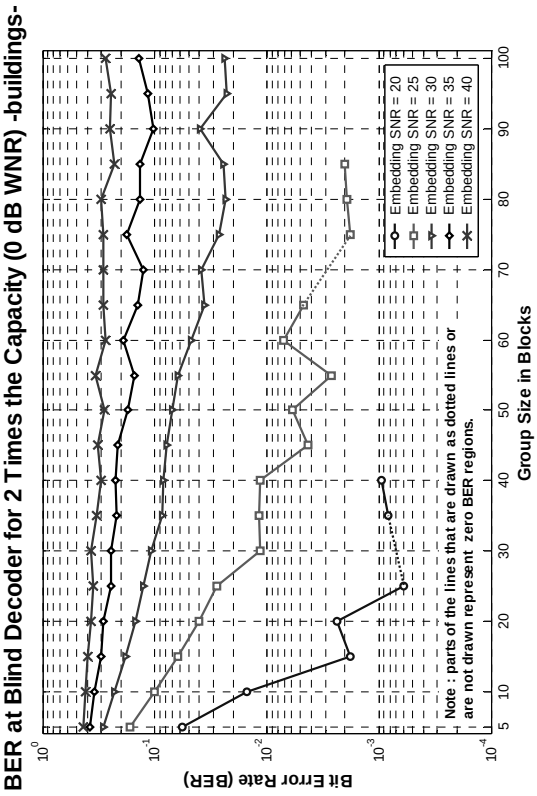


Fig. 4.18 (d)

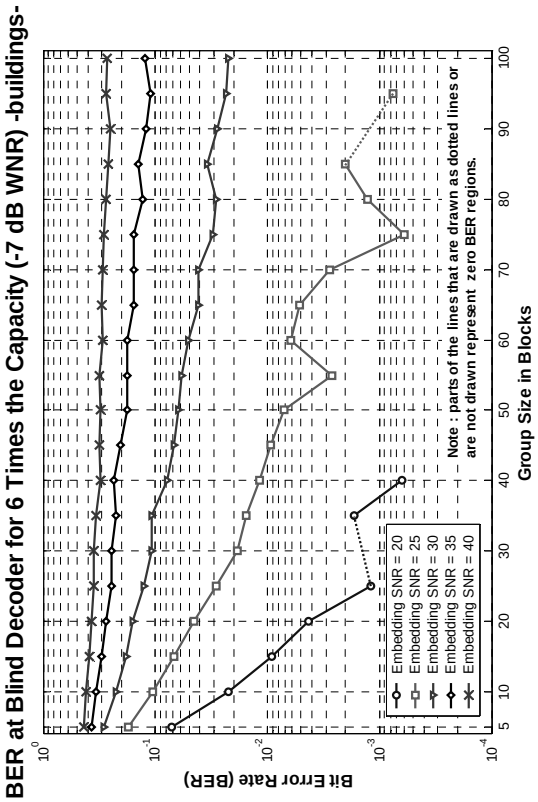


Fig. 4.18 (e)

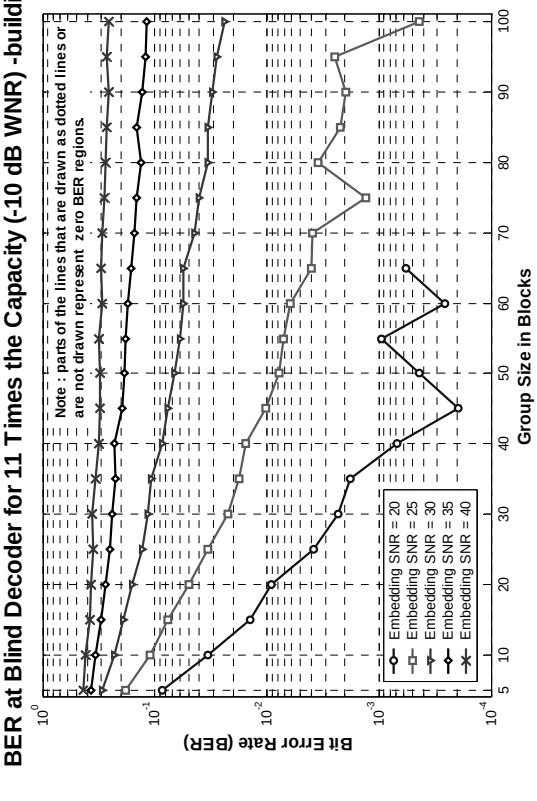


Fig. 4.18 (f)

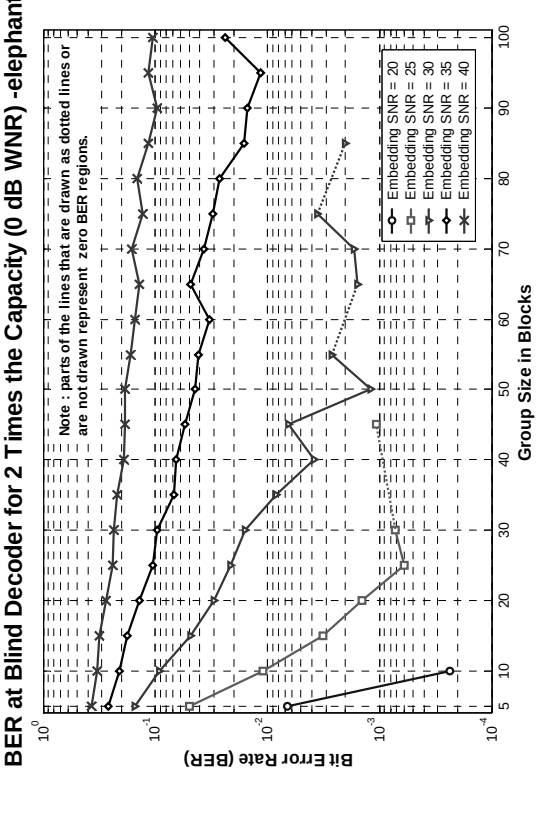


Fig. 4.18 (g)

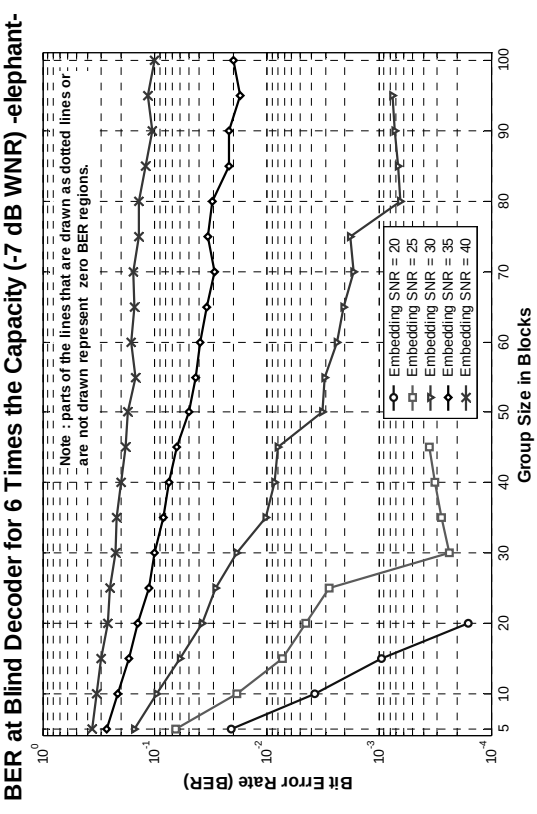


Fig. 4.18 (h)

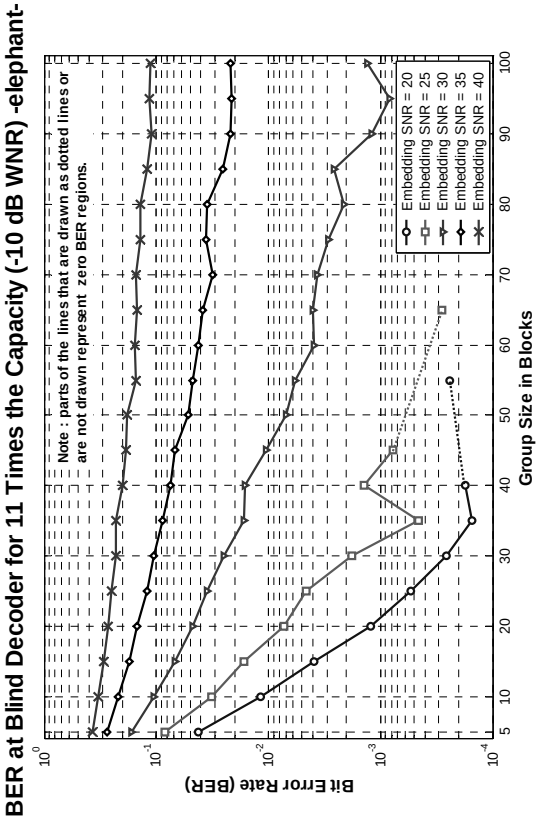


Fig. 4.18 (i)

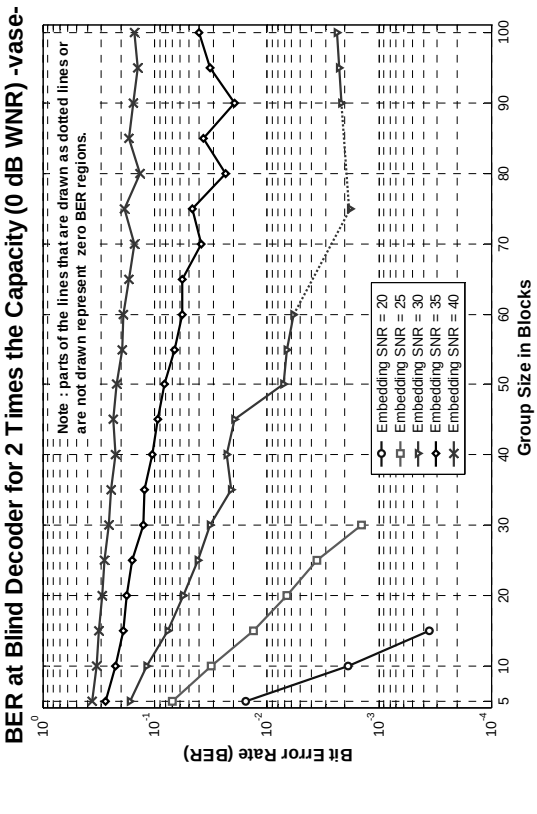


Fig. 4.18 (j)

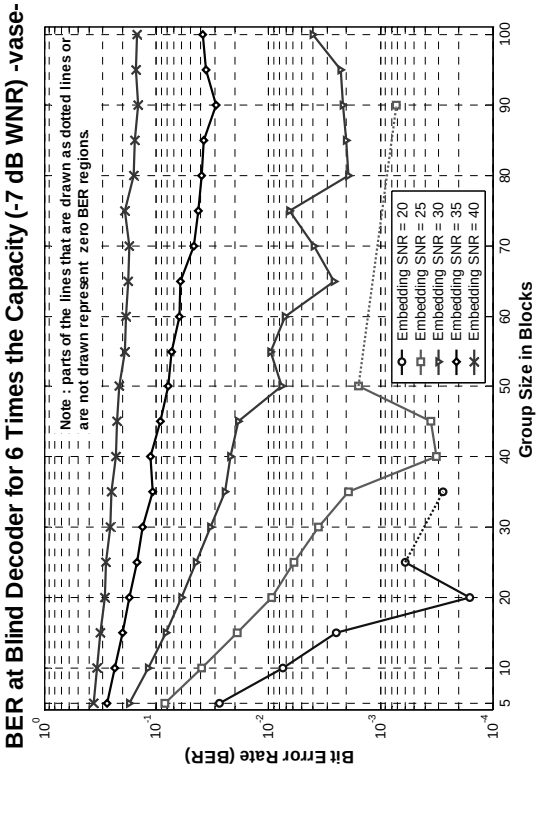


Fig. 4.18 (k)

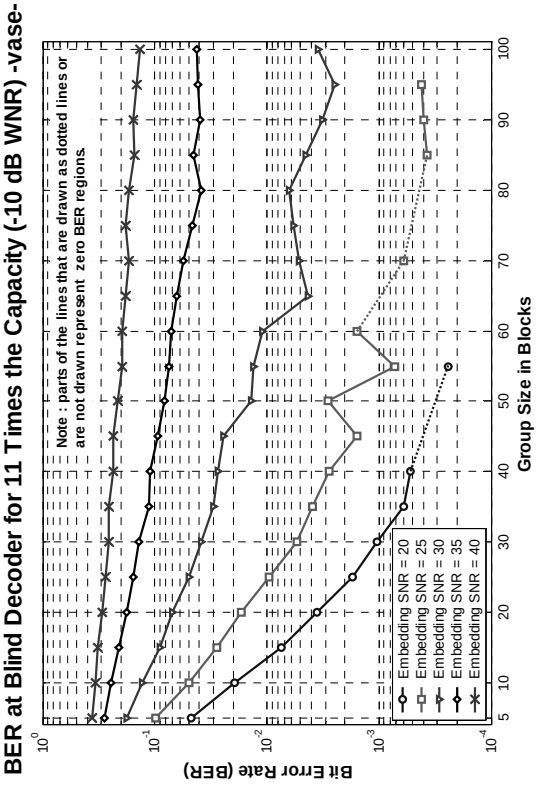


Fig. 4.18 (l)

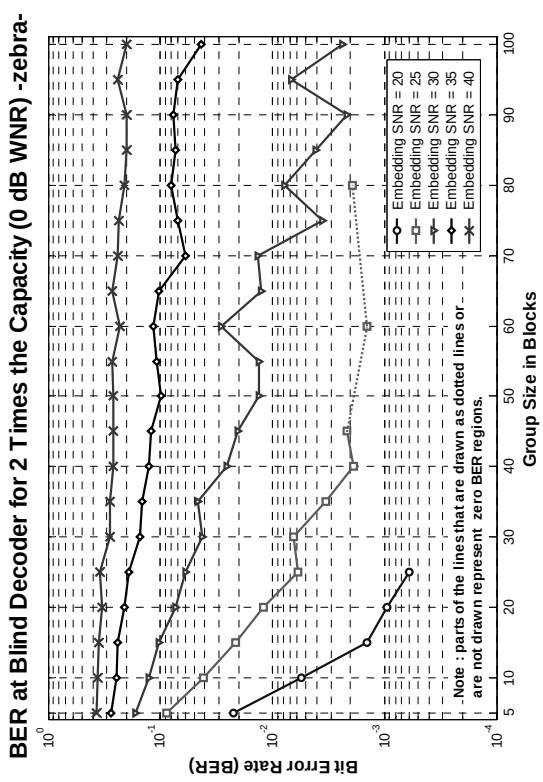


Fig. 4.18 (m)

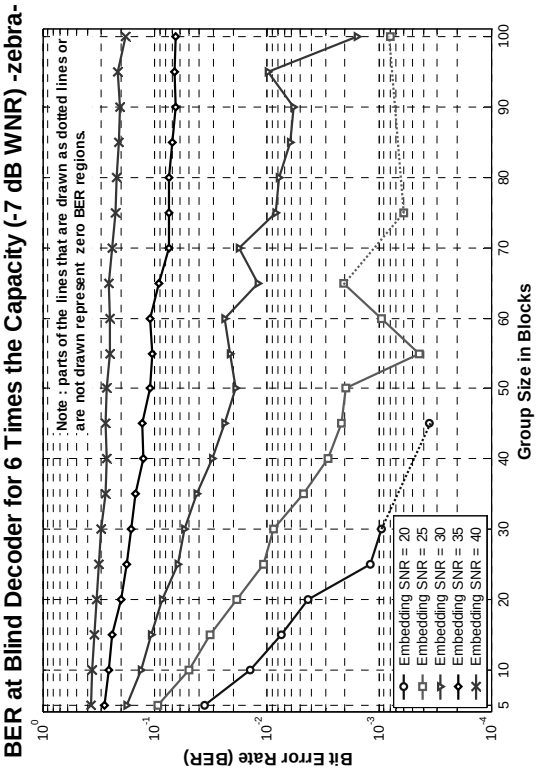


Fig. 4.18 (n)

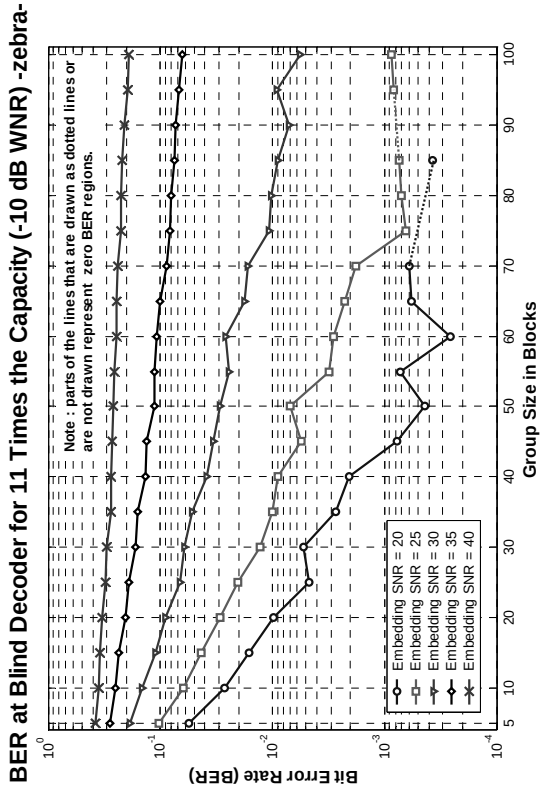


Fig. 4.18 (o)

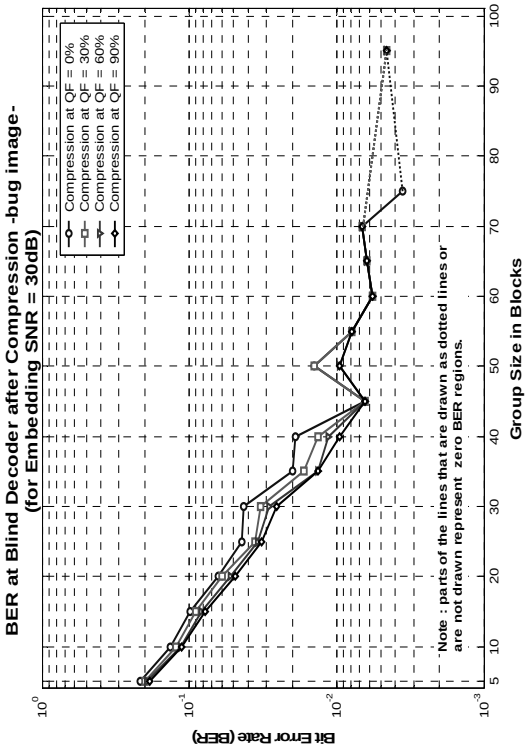


Fig. 4.19 (a)

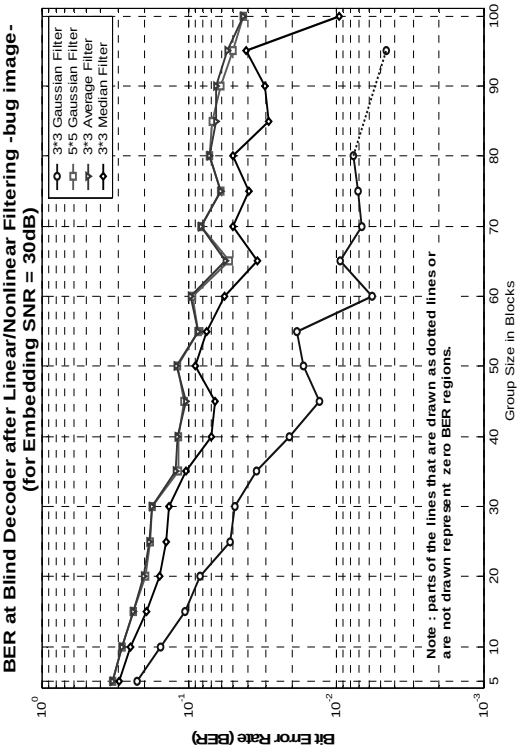


Fig. 4.19 (b)

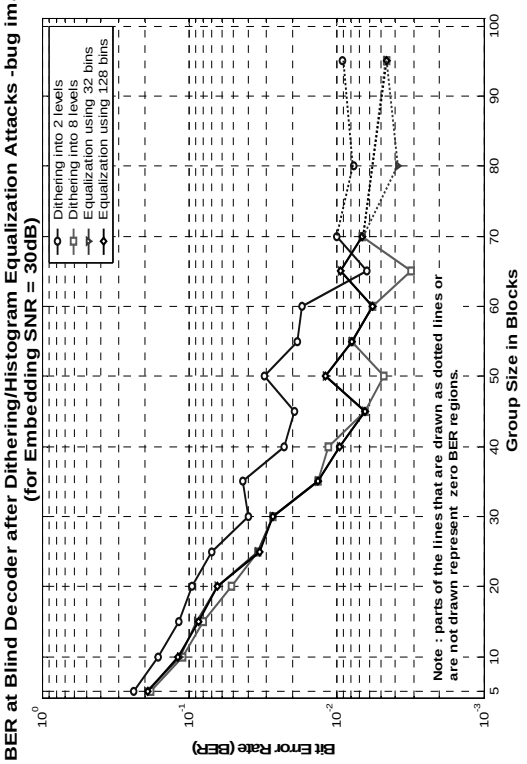


Fig. 4.19 (c)

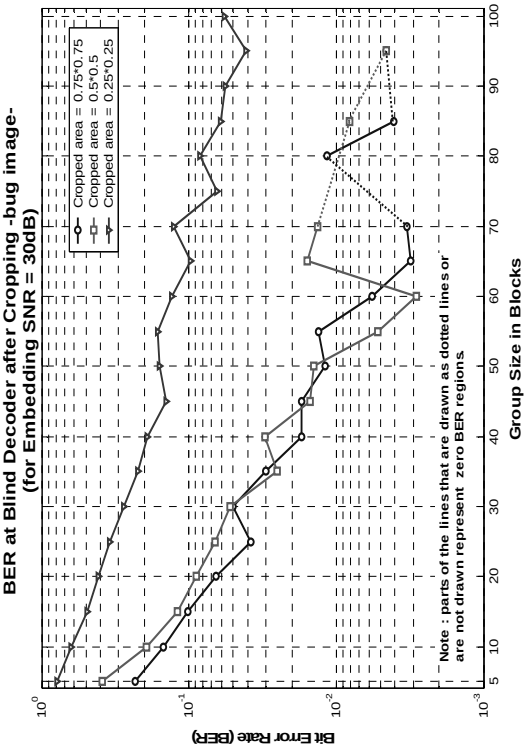


Fig. 4.19 (d)

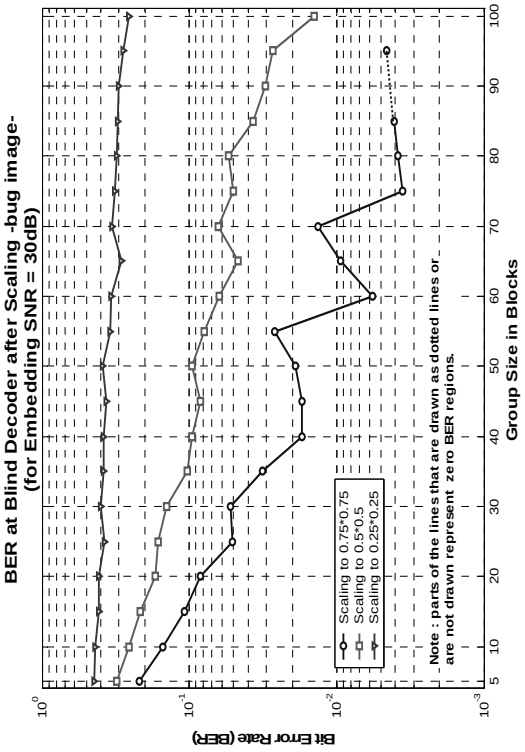


Fig. 4.19 (e)

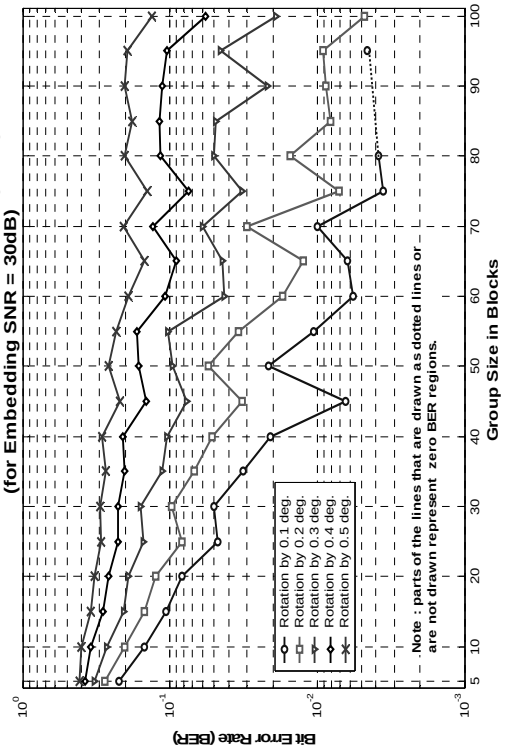


Fig. 4.19 (f)

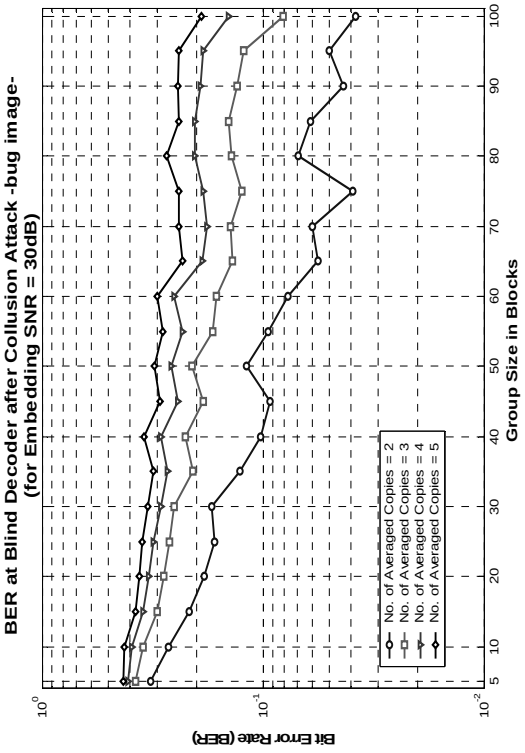


Fig. 4.19 (g)

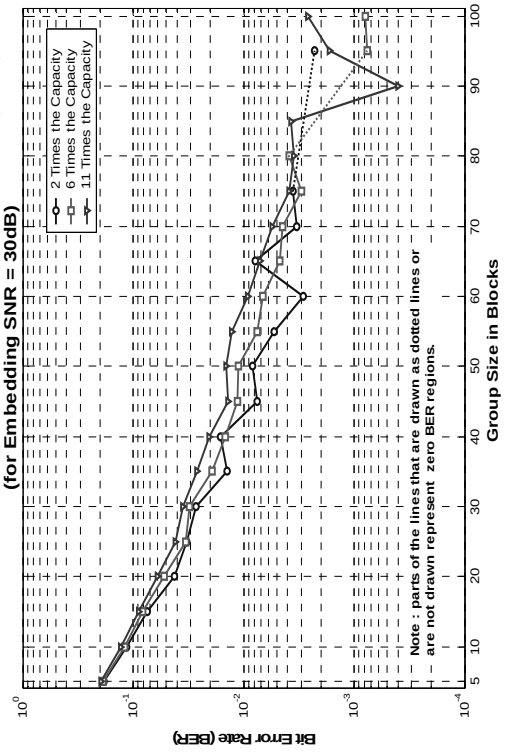


Fig. 4.19 (h)

CHAPTER FIVE

Conclusions and Future Work

5.1 Conclusions:

1. In the course of the work here, a modified spread spectrum watermarking system has been developed, which can generally be viewed as the hierarchical combination of both traditional spread spectrum modulation (encoding) techniques and integrate-and-dump preprocessing.
2. As it has been proved both theoretically and experimentally, the proposed modifications here do not impose any degradation on the robustness of the embedded data (in comparison to the equivalent traditional spread spectrum systems) nor -practically- compromise its security.
3. The new implementation of redundancy -proposed by this modified system- and the way it is processed here offers an opportunity to apply some form of a relative host-adaptation scheme that is shown to induce no actual distortions into the embedded data signal, which eliminates the need for any adaptation distortion equalization processing prior to decoding. This enables the encoder to freely select among the available different perceptual adaptation models or even adopt a new model without having to notify the decoding side or to update the decoding system.
4. An important advantage of the proposed watermarking system over the traditional spread spectrum watermarking systems is the efficient replacement of the time consuming array multiplication spreading processing with the much simpler accumulation processing here. This efficiency gain is proportional to the integrate-and-dump to spread spectrum processing tradeoff. On the other side, bringing the adaptation equalization processing into the encoding side can greatly

simplify the structure of the corresponding decoding systems and help *real-time* decoding.

5.2 Suggestions for Future Work:

1. In the future, a more sophisticated host-perceptual-adaptation model is suggested to be applied. This -generally- is expected to further enhance the perceptual quality of the system output without imposing any degradation to the embedding robustness, as is implied by the proposed host-adaptation scheme.
2. To achieve different robustness-imperceptibility characteristics using the proposed modified spread spectrum watermarking scheme, different watermark embedding domains can be employed, such as the wavelet transformation domain.
3. The geometrical invariance properties of the proposed watermarking technique here can be enhanced through the employment of the so-called geometrically invariant watermark mapping techniques. This can prove to be a significant countermeasure against geometrical and desynchronization attacks.

APPENDIX A

Notation Conventions

Notation Conventions:

The notation below will be used throughout this thesis:

Scalar random variables will be denoted by uppercase letters (e.g. X), with their individual values being denoted by italicized lowercase letters (e.g. x). Probability distributions of these variables will be denoted by $p_X(x)$ (or simply $p(x)$ when used as subscripts). On the other hand, bold letters (e.g. $\mathbf{X}$, $\boldsymbol{x}$, $p_{\mathbf{X}}(\boldsymbol{x})$) will be used with vector random variables. Vector random variables will also be denoted by superscripted letters (e.g. X^N, x^N, $p_{\mathbf{X}}(x^N)$) to emphasize on their dimensionality, where $\mathbf{X} = X^N = (X_1, X_2, ..., X_N)$ is a block of N independent scalar random variables (where $X_i \in \mathcal{X}$, for i = 1, 2, … N).

Sets of numbers will be denoted by script letters (e.g. $\mathcal{X}$ is the set of all values that can be assumed by the variable X, or simply $X \in \mathcal{X}$). The cardinality of these sets will be denoted by | . | (e.g. $|\mathcal{X}|$ is the number of elements within the set $\mathcal{X}$). Script letters with superscripts will be used to represent Cartesian products of sets (e.g. $\mathcal{X}^N = \mathcal{X} * \mathcal{X} * \ldots * \mathcal{X}$).

Constant values (e.g. the dimensionality of some vector N) will be denoted here by uppercase letters. These symbols, however, will not be confused with scalar variables, as will be seen in the corresponding contexts.

Subscripts -on the other hand- will be used to specify the different dimensions of vector variables or the columns of arrays (e.g. $X^N = (X_1, X_2, ..., X_N)$). Note here that subscript indexes will be used within the elements of some sets where emphasis on the distinction (or selection) among them is necessary. It is emphasized here though that these indexes are not aimed at proposing any specific ordering among these elements.

The notation I(X;Y) will be used to represent the mutual information between random variables X and Y, while I(X;Y | Z) will be used for

conditional mutual information with condition Z. Joint and conditional probability distributions will be represented by the notations $p_{X,Y}(X, Y)$ and $p_{X \mid Y}(X \mid Y)$, respectively, while attack channels will be described by the conditional probability distribution $A^N(y^N \mid x^N)$ for some input vector X^N and output vector Y^N.

Continuous and discrete time signals (functions) will take the usual functional form with parameters t and n to represent the continuous and discrete time variables, respectively, (e.g. r(t), r(n)), while deterministic functions with vector inputs and outputs will be denoted by lowercase letters with subscript N (e.g. $y^N = g_N(x^N)$).

The dot operator " $\cdot$ " will be used to represent component-by-component products of vectors (e.g. $\boldsymbol{x} \cdot \boldsymbol{y}$), while inner products (or correlation coefficients) will be denoted by $< \boldsymbol{x} , \boldsymbol{y} >$.

REFERENCES

References

[1] I. J. Cox and M. L. Miller, "Electronic watermarking: the first 50 years", *Proceedings of IEEE, Workshop on Multimedia Signal Processing*, 2001.

[2] F. A. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information hiding – a survey", *Proceedings of IEEE, special issue on protection of multimedia content*, vol. 87, No. 7, pp. 1062-1078, July 1999.

[3] N. F. Johnson, Z. Duric and S. Jajodia, "Information hiding: steganography and watermarking – attacks and countermeasures", *Center for Secure Information Systems, George Mason University, Kluwer Academic Publishers*, 2001.

[4] N. F. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen", *IEEE Computing Practices*, pp. 26-34, February 1998.

[5] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies", *Proceedings of IEEE*, vol. 86, No. 6, pp. 1064-1087, June 1998.

[6] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding", *IEEE Transactions on Information Theory*, vol. 49, No. 3, pp. 563-593, March 2003.

[7] A. Skodras, C. Christopoulos and T. Ebrahimi, "The JPEG 2000 still image compression standard", *IEEE Signal Processing Magazine*, pp. 36-58, September 2001.

[8] R. B. Wolfgang, C. I. Podilchuk and E. J. Delp, "Perceptual watermarks for digital images and video", *Video and Image Processing Laboratory* (*VIPER*), *School of Electrical and Computer Engineering, Purdue University*. Address all correspondence to E. J. Delp at ace@ecn.purdue.edu, or http://www.ece.purdue.edu/~ace.

[9] J. Cummins, P. Diskin, S. Lau and R. Parlett, "Steganography and digital watermarking", *School of Computer Science, University of Birmingham*, Copyright © 2004.

[10] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography", *IEEE Journal of Selected Areas in Communications, Special Issue on Copyright & Privacy Protection*, vol. 16, No. 4, pp. 474-482, May 1998.

[11] A. Menezes, P. van Oorschot and S. Vanstone, "Handbook of applied cryptography", *CRC Press*, 1997.

[12] J. Fridrich, "Secure encryption and hiding of intelligence data", *Air Force Research Laboratory, Information Directorate, Rome Research Site, Rome, New York*, Final technical report, AFRL-IF-RS-TR-1998-192, September 1998.

[13] H. H. Al-Obaidi, "Encryption using wavelet coded image data", M.Sc. Thesis, *Dept. of Computer Engineering, College of Engineering, University of Basrah*, June 2004.

[14] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure spread spectrum watermarking for multimedia", *IEEE Transactions on Image Processing*, vol. 6, No. 12, pp. 1673-1687, 1997.

[15] W. Trappe, M. Wu, Z. J. Wang and K. J. Liu, "Anti-collusion fingerprinting for multimedia", *IEEE Transactions on Signal Processing*, vol. 51, No. 4, pp. 1069-1087, April 2003.

[16] M. Wu, W. Trappe, Z. J. Wang and K. J. Liu, "Collusion-resistant fingerprinting for multimedia", *IEEE Signal Processing Magazine*, pp. 15-27, March 2004.

[17] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction", *IEEE Transactions on Image Processing*, vol. 13, No. 12, pp. 1627-1639, December 2004.

[18] Y. K. Lee and L. H. Chen, "High capacity image steganographic model", *IEE Proceedings – Vis. Image Signal Processing*, vol. 147, No. 3, June 2000.

[19] J. Zhao and E. Koch, "Embedding robust labels into images for copyright protection", *Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Vienna, August 1995.

[20] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding", *IBM Systems Journal*, vol. 35, No. 3&4, pp. 313-336, 1996.

[21] L. M. Marvel, C. G. Boncelet Jr. and C. T. Retter, "Reliable blind information hiding for images", *David Aucsmith (Ed.): Information Hiding*, LNCS 1525, pp. 48-61, Copyright © Springer-Veriag Berlin Heidelberg, 1998.

[22] J. R. Hernández and F. Pérez-González, "Statistical analysis of watermarking schemes for copyright protection of images", *Proceedings of IEEE*, vol. 87, No. 7, pp. 1142-1166, July 1999.

[23] B. Chen, "Design and analysis of digital watermarking, information embedding and data hiding systems", Ph.D. Thesis, *Dept. of Electrical Engineering and Computer Science*, *Massachusetts Institute of Technology*, June 2000.

[24] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding", *IEEE Transaction on Information Theory*, vol. 47, No. 4, pp. 1423-1443, May 2001.

[25] B. Chen and G. W. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia", *Journal of VLSI Signal Processing* 27, pp. 7-33, *Kluwer Academic Publishers*, 2001.

[26] C. Fei, D. Kundur and R. H. Kwong, "Analysis and design of watermarking algorithms for improved resistance to compression", *IEEE Transactions on Image Processing*, vol. 13, No. 2, pp. 126-144, February 2004.

[27] R. L. Pickholtz, D. L. Schilling and L. B. Milstein, "Theory of spread-spectrum communications – a tutorial", *IEEE Transactions on Communications*, vol. 30, No. 5, pp. 855-884, May 1982.

[28] E. Ström, T. Ottosson and A. Svensson, "An introduction to spread spectrum systems", *Dept. of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden*, Technical report No. R016/2002, second corrected printing, January 7th 2004.

[29] H. Taub and D. L. Schilling, "Principles of communication systems", *McGraw-Hill Kogakusha, Ltd.*, 1971.

[30] C. L. Liu, "Elements of discrete mathematics", *McGraw-Hill Book Company*, second edition, 1987.

[31] S. W Smith, "The scientist and engineer's guide to digital signal processing", *California Technical Publishing*, second edition, 1999.

[32] A. B. Watson and J. A. Solomon, "Model of visual contrast gain control and pattern masking", *Optical Society of America*, vol. 14, No. 9, pp. 2379-2391, September 1997.

[33] J. F. Delaigle, C. Devleeschouwer, B. Macq and I. Langendijk, "Human visual system features enabling watermarking", this paper has been submitted as a part of a *Proceedings of IEEE paper*, E-mail: {devlees,delaigle,macq}@tele.ucl.ac.be.

[34] "Image processing toolbox user's guide", *MathWorks Inc.*, version three, July 2002.

الخلاصة

نتيجة لاتساع نطاق التعامل مع الوسائط المتعددة الرقمية (digital multimedia) و ازدياد سهولة التلاعب بها و تناقلها، فإن حماية هذه الوسائط من الإستخدام غير القانوني لها قد أصبح يشكل تحديا حقيقيا. لقد جوبهت هذه المشكلة على مدى طويل من خلال استخدام تقنيات التشفير (encryption) لحظر الوصول إلى محتويات هذه الوسائط بشكل غير مصرح به، غير أن تقنيات التشفير المستخدمة تصبح عاجزة عن حماية محتويات هذه الوسائط بمجرد أن يتم فك شفرتها. و لذلك اقتُرح لمواجهة هذا التحدي أن يتم تضمين بعض المعلومات التي تثبت ملكية حقوق النشر لأصحابها (و التي أصبحت تسمى العلامة المائية -watermark-) في محتويات هذه الوسائط بشكل دائم.

في هذه الأطروحة، تم اقتراح نظام لتضمين العلامة المائية باستخدام تقنية معدلة من الطيف المنتشر (spread spectrum). النظام المقترح هنا يمتاز عن أنظمة الطيف المنتشر التقليدية الجاسئة بكونه أكثر حبيبية، حيث أنه يجزئ ميكانيكية انتشار الطيف التقليدية إلى عدد من الميكانيكيات المكررة و الأقل أبعادا (dimensions)، مع تعويض انخفاض كسب المعالجة (processing gain) الناتج عن تخفيض أبعاد ميكانيكيات انتشار الطيف من خلال استخدام تقنية المراكمة- و- الإخراج (integrate-and-dump) لاستغلال التكرار المتوفر.

و لتحسين أداء النظام المقترح من حيث الجودة الحسية (perceptual quality) لإخراجه، تم اقتراح نمط جديد لتكييفه للمضيف (host-adapting) بما يضمن إعادة توزيع التشويه الحسي (perceptual distortion) -المستحث من قبل العلامة المائية المضمنة- بشكل متساوٍ (uniformly) خلال المضيف و بدون أن تتسبب ميكانيكية التكييف هذه بأي تشويهات للعلامة المائية المضمنة نفسها، مما يعني عدم الحاجة إلى أي معالجة لإزالة تشويهات التكييف (adaptation distortion equalization processing) من قبل الجهة المستلمة (receiving end).

تضمين العلامة المائية المتكيفة للمضيِّف
باستخدام تقنية معدلة من
الطيف المنتشر

رسالة ماجستير مقدمة الى

كلية الهندسة - جامعة البصرة

كجزء من متطلبات نيل درجة

ماجستير علوم في هندسة الحاسبات

من قبل

علي عصام حميد حداد

بكالوريوس، جامعة البصرة (٢٠٠٣)

بإشراف

الدكتور
ماجد عبد النبي علوان

كانون الثاني ٢٠٠٦