We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

15

An Application of Digital Image Restoration Techniques to Error Control Coding

Pål Ellingsen
Department of Computer Science
University College of Bergen
Norway

1. Introduction

Digital image restoration has been a field of very active research for many years, and digital image restoration techniques has been put to use in a lot of different contexts including astronomy, medicine, intelligence work and many others (Banham & Katsaggelos (1997)). Common to these fields of application is that the restoration techniques are applied to image data of some kind true to the original intentions of the algorithms. In this text we present an application of principles from digital image restoration to the field of coding theory, and the objects of application are not images but rather general information data.

Information can be represented in many different ways. A typical approach in information theory is to represent information as binary vectors, but there are many situations where information can rather be represented as a matrix or grid containing the information symbols giving rise to the concept of two-dimensional channels. Good examples of this can be found in the fields of magnetic and optical storage, bar codes and others.

When transmitting information of any kind, a central problem is how to deal with errors that result from the transmission process, and a solution to this problem is to add redundancy to the information in such a way that it is possible to detect and eventually also correct the errors that occur. Adding this redundancy is called error control coding, and the techniques for doing so is called error correcting or detecting codes. There exists a huge variety of error control coding techniques for channels with different characteristics and for fulfilling different sets of requirements. However, most of the channel coding techniques assumes the information that is to be encoded, are one-dimensional vectors or a stream of information symbols. Channel coding for two-dimensional channels on the other hand, is a part of coding theory that has only recently attracted attention from the coding theory community.

There are different models for describing how errors occur in a two-dimensional communication channel.

- The errors can be modeled as independent and identically distributed over the information symbols. In this case the problem of error control coding is reduced to the case of error control coding for a one-dimensional channel with an equal information rate.
- The errors can be modeled by considering two-dimensional intersymbol interference, which is the effect of the information symbols interfering with their neighboring information symbols. This error model applies to many practical communication channels, most notably magnetic and optical storage media (which can be seen as

communication channels). This error model has been studied extensively, see for example Singla & O'Sullivan (2005) and Kurkoski (2008).

- The errors can be modeled as spatially contagious areas bounding a cluster of errors. The underlying channel model assumes that the information symbols are affected by some physical process that affects a limited part of the information. Error that result from such processes would form error bursts that may take the form of clusters or be concentrated to a limited area. Such error clusters are defined differently in the literature, but a very common approach is to define an error cluster as a rectangular area of a given size $n_1 \times n_2$. Code constructions for this type of cluster errors can be found in Farrell (1982) Schwartz & Etzion (2005) and Breitbach et al. (1998). More recently, error clusters of arbitrary form has been considered in several works, and most of these use interleaving strategies to correct cluster errors. This approach is used in Blaum et al. (1998), Schwartz & Etzion (2003) and Xu & Golomb (2007).

In the following we take the latter perspective on the nature of errors on a two-dimensional channel, and we apply techniques from digital image restoration to support the decoding process since these methods can exploit the information inherent in the two-dimensional cluster error model. The core element in this application of digital image restoration is looking at the encoded information as "noise" and the areas affected by burst errors as the "original image" that we want to restore. The strategy is to first encode the information at the source using ordinary error control coding. At the receiver, ordinary decoding and detection of the error clusters are combined in an iterative system where the decoding process produces an estimate of the probability of error in each information symbol, and this estimate is then used as input to a component that produces an estimate of the size and shape of the error burst based on image restoration techniques. The information that is extracted from this process is then used to support the decoding process as a priori information about the error clusters.

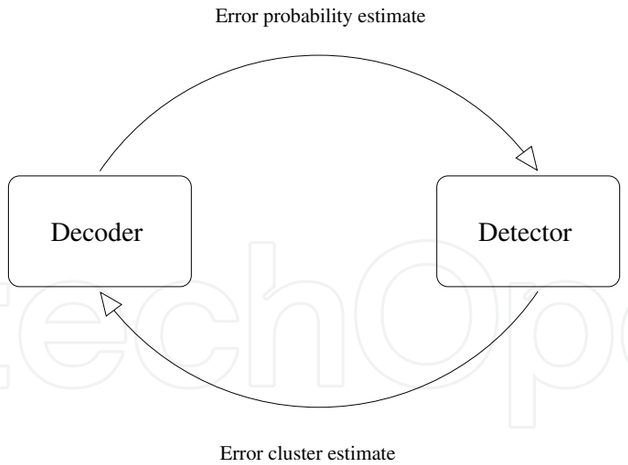


Fig. 1. Basic principle of iterative process

2. General overview over relevant image restoration techniques

In our application we are concerned with describing the statistical properties of context dependent entities such as neighboring bits in a two-dimensional representation of digital information. One technique for describing such properties is the use of Markov random field theory which uses conditional probabilities to describe spatial dependencies in an n-dimensional system. This approach is based on the results of Shridhar, Ahmadi and

El-Gabali who developed the applied techniques in Shridhar et al. (1989), El-Gabali et al. (1987), El-Gabali et al. (1988) and El-Gabali et al. (1990) but similar techniques are also presented in Geman & Geman (1993), Zhang (1993), Jeng & Woods (1991) and Chalmond (1988). The basis for all of these image restoration techniques is that simulated annealing is used to produce an estimate of the maximum a posteriori probability of the original image and the techniques has been extensively used for different purposes within the field of image processing, including image restoration, image segmentation, object identification and texture analysis. Using a system model based on Markov random field theory, one wants to find the joint distribution of the variables representing the image (e.g. pixels) and then use this distribution as basis for detecting the original scene and eliminate or reduce noise in the image. However, finding the joint distribution directly from a Markov random field model is mathematically intractable, so one needs to compute the distribution by aid of the so-called Hammersley-Clifford theorem which states the equivalence between the joint distribution of the variables in a Markov random field and Gibbs distribution which can be treated mathematically in an efficient way. Given this distribution one common method for performing the actual detection of the original image is to find the maximum a priori probability for the image given the observed output.

3. General overview of the iterative decoding and detection process

two-dimensional channels are subjected to different kinds of errors, but in this setting we are interested in sources of errors that will affect spatially limited parts of a two-dimensional codeword. This is called an error cluster or equivalently a burst error. Burst error correction is a well known and much studied problem, but none of the classical techniques in this field are able to take into account the fact that such spatially correlated error clusters gives rise to a statistical correlation on the error probability for neighboring positions in the codeword. Using the above mentioned techniques from digital image restoration is one way one can exploit this extra information given in the error model.

Several different approaches are possible when trying to use the information gleaned from the image restoration to enhance the decoding process. However, our approach is based on the use of so-called soft input - soft output (SISO) decoding. The principle behind this decoding strategy is that the decoder should accept input values in the form of conditional probabilities as a measure of the reliability of the corresponding channel value, and as output produce a new measure of reliability for the corresponding channel value. Such a decoder can take advantage of the information produced by the restoration process in a very natural way by supplying conditional probabilities resulting from the estimation process described below.

Based on results from our papers Ellingsen et al. (2004) Ellingsen & Kvamme (2010) we show how this principle can be used to implement an actual decoding system based on the techniques described above. We study two different channel models, the two-dimensional binary asymmetric channel and the two-dimensional binary symmetric channel, and use LDPC-codes for error correction. Then we show how the redundant information of the code can be used to provide prior information to an image restoration process, while the results from the image restoration process is used to assist the decoding process by providing a priori information to the decoder. Thus we construct an iterative decoding process where estimates in the form of conditional probabilities for each bit in the codeword is exchanged back and forth between the LDPC-decoder and the image restoration module. The results from this process are compared to the results when using the LDPC decoder alone and we see that there is a significant gain in performance.

4. Details of restoration technique

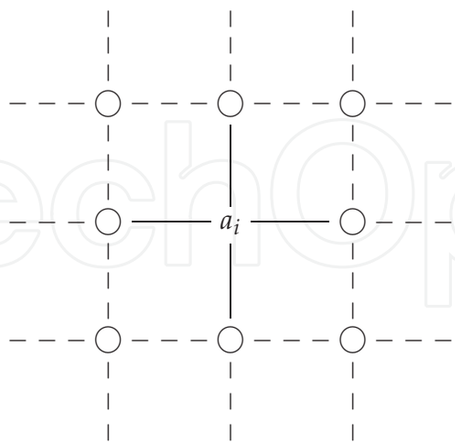
4.1 Modelling a two-dimensional channel using Markov Random Fields

A channel with memory is characterized by the existence of dependencies in the noise generating process. Such dependencies can e.g. be described by a Markov chain in the case of one-dimensional channels as is the case for the Gilbert-Elliott channel Gilbert et al. (1960). This implies that the channel will have characteristics that are varying with time. We want to extend this line of thinking to the case of two-dimensional channels and look at spatial dependencies in the noise generating process rather than temporal dependencies as in the one-dimensional case. Such spatial dependencies can be modeled using a Markov Random Field (MRF).

An MRF can be seen as a generalization of Markov chains, but while a Markov chain is often defined over a domain of time as a sequence of random variables, an MRF can be defined in space to describe dependencies between variables on a grid of dimension 2 or higher.

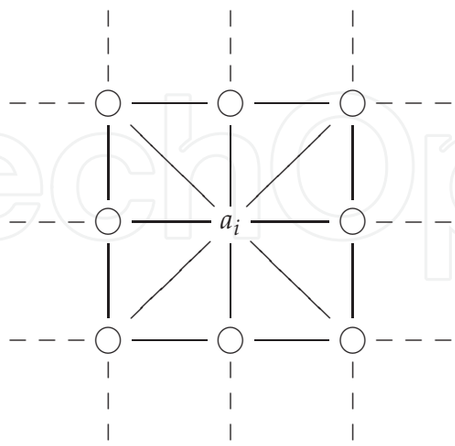
4.2 Markov Random Fields

Consider a set of random variables $A = \{A_i | i \in I\}$ for some index set I, where the variables are organized in a two dimensional grid. Let the variables correspond to the vertices and the statistical dependencies between the variables correspond to edges in an undirected graph G. We shall use this setup to model both codewords and errors in our system. Two connected vertices in G are said to be *neighbors*, and a *neighborhood* $\mathcal{N}_i$ of a vertex a_i can be defined as the set of vertices that are connected to a_i in G. Different sizes of neighborhoods can be defined for an MRF. By convention, a node is not a neighbor of itself. On a regular lattice we define the first order neighborhood to be the four closest neighbors of a node as seen below, the second order neighborhood as the eight closest neighbors and so on. The collection of all neighborhoods $\mathcal{N} = \{\mathcal{N}_i \mid \forall i \in I\}$ in a graph, is called a *neighborhood system*.

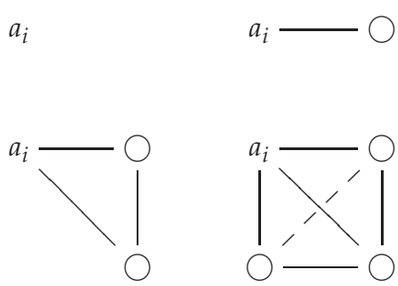


Within the neighborhood of a vertex a_i, we define a *clique* to be any collection of vertices that contains a_i and forms a fully connected subgraph of G, i. e. that the vertices are mutual neighbors relative to the neighborhood system $\mathcal{N}$. In the case of a first order neighborhood, all nodes within distance 1 of the center are said to be neighbors, and the cliques become

the center node a_i and all pairs of (a_i, a_j) where a_j is a neighbor of a_i. In a second order neighborhood, all nodes within distance $\sqrt{2}$ are defined as neighbors:



and in this case the cliques becomes any configuration of



The collection of all cliques of size i in a neighborhood system $\mathcal{N}$ is called $\mathcal{C}_i$. The set $\mathcal{C}$ of all cliques in a graph can then be partitioned into the subsets $\mathcal{C}_i$ for $1 \leq i \leq n$

Now, based on the concept of neighborhoods we can then proceed to define a Markov Random Field. Just as a Markov chain $\{\ldots, a_k, a_{k-1}, a_{k-2}, \ldots\}$ satisfies

$$P(a_i|a_{i-1}, a_{i-2}, \ldots) = P(a_i|a_{i-1}, a_{i-2}, \ldots, a_{i-n})$$

for some n, a Markov Random Field should satisfy

$$P(a_i|a_{I-\{i\}}) = P(a_i \mid \mathcal{N}_i)$$

where I is the set of indices of a and $\mathcal{N}_i$ is the neighborhood of a_i as defined above.

4.3 Probability distribution

The fact that the errors of our channel can be represented by an MRF does not immediately enable us to analyze the error patterns statistically. By assuming that the dependencies in a collection of random variables can be represented by an MRF, the joint probability of the variables is given by the so called *Gibbs distribution*.

Definition 1 (Gibbs distribution). *A set of random variables is said to be a* Gibbs random field *(GRF) if the joint distribution of the variables takes the following form:*

$$P(X = x) = \frac{1}{Z} \exp\left[-\frac{1}{T} U(X)\right] \tag{1}$$

This distribution is called a Gibbs distribution.

- Z is a constant called *the partition function* and can be expressed as $Z = \sum_{x \in \mathbb{X}} e^{-\frac{1}{T}U(x)}$, so that Z^{-1} becomes a normalizing constant in the expression.
- $U(x)$ is called *the energy function* and is a function of the values of the variables forming cliques in the field. It can be written as

$$U(x) = \sum_{c \in \mathcal{C}} V_c(x) \tag{2}$$

  We can expand (2) further by summing over the cliques of the same degree separately

$$\sum_{c \in \mathcal{C}} V_c(x) = \sum_{a \in \mathcal{C}_1} V_1(a) + \sum_{a,b \in \mathcal{C}_2} V_2(a,b) + \sum_{a,b,c \in \mathcal{C}_3} V_3(a,b,c) + \dots \tag{3}$$

  where $\mathcal{C}_i$ is the collection of all cliques of degree i, so that V_i is a function of i variables forming a clique, and $\sum_{\mathcal{C}_i} V_i$ mean that we sum over all possible cliques in the field of degree i.
- T is called the *temperature* (this is a legacy from the distribution's origin in statistical physics). The parameter T influences the degree of cohesion between the variables on a grid, so that a higher temperature corresponds to a lower degree of cohesion in the sense that the values of the variables becomes more and more independent, while a lower temperature gives a higher probability of the formation of large clusters of variables with the same value. We shall assume that the temperature is 1 in our simulations, even if the parameter will be used in the theoretical treatment of the decoding algorithm.

The Clifford-Hammersley theorem states that for a set of variables $\mathcal{F}$ with a neighborhood system $\mathcal{N}$, $\mathcal{F}$ is an MRF with respect to $\mathcal{N}$ if and only if $\mathcal{F}$ is a GRF with respect to $\mathcal{N}$. See Kindermann & Snell (1980).

Unfortunately, Z is very hard to compute. Since we have to consider all possible of values of x in order to find Z, the computational complexity of the task is a formidable $O(2^n)$, effectively preventing us from computing the absolute probabilities for the configurations of X. It is nevertheless possible to use the Gibbs distribution to find an estimate of the error patterns generated by the channel.

4.4 MAP estimation

We want to find an estimate of the error pattern that was added to the codeword, based on the assumptions about the dependence between errors given in the previous sections. In order to avoid computing the constant Z in the Gibbs distribution, we will do a MAP estimation of the errors. That is, given a received word Y, we want to find an estimate of the most likely error pattern X that was added to C. Some terminology is needed in order to develop this.

Let A be a set of random variables defined on the set $\mathcal{L}$, and let the elements of A be indexed by $1 \leq i \leq n$. If $A_i = a_i$ for each variable A_i, where $a_i \in \mathcal{L}$, we call $\{a_1, \dots, a_n\} = a$ a *configuration* of A

MAP estimation of the error pattern X based on the received word Y can be formulated as the optimization of the *a posteriori* probability $P(X = x|Y = y)$ with respect to x. That is, we want to find a configuration x that makes the probability $P(X = x|Y = y)$ as high as possible.

Bayes rule gives us

$$P(X = x|Y = y) = \frac{P(X = x)P(Y = y|X = x)}{P(Y = y)}$$

Since $P(Y = y)$ does not depend on $P(X = x)$, we can maximize over

$$P(X = x)P(Y = y|X = x) \tag{4}$$

To find the probabilities $P(Y = y|X = x)$, we must take care to remember that the error pattern X is now considered as the original information that we want to estimate, and the codeword C is to be considered as errors obscuring the information. In the following, we shall make some assumptions about X and C.

- The variables are bipolar, with 1 corresponding to 0 and -1 corresponding to 1 in the channel model.
- The codeword C, when treated as errors, can be seen as random bipolar variables so that

$$P(C = c) = \prod_i P(c_i) = (\frac{1}{2})^n$$

The conditional probabilities must then be expressed by by using the characteristics of the channel and this expression must then be optimized with respect to the input configuration. In chapter 5.3 and 6.3 we show two examples of how this optimization can be done for a given channel. In both of our cases, finding a global maximum of the conditional probabilities with respect to x would become computationally infeasible as the size of x increases. Instead, we use the local dependencies between bits to do a local optimization along the lines of the PDFE in Neifeld & King (1998); Neifield et al. (1996) or the partial binary segmentation algorithm of Shridhar et al. (1989).

4.5 Error generation

Generation of two dimensional burst errors for simulation purposes is done by the use of a Monte Carlo Markov chain technique called the Metropolis algorithm. We do not have very strict requirements for the generated sample configurations, other than that they should be "somewhat likely" to occur given the condition that the variables' distribution is given by the Gibbs distribution.

The Metropolis algorithm is a general method for generating samples from a joint distribution of two or more variables, and can be applied to distributions that are either continuous or discrete as long as it is possible to compute the difference of the likelihoods for two configurations of the variables.

We would like to sample the joint distribution $A = \{A_1, \ldots, A_n\}$. This is achieved by generating random changes to the components A_i of A, and accepting or rejecting these changes based on how they affect the likelihood of the configuration. In our case, the natural change to a component of a configuration would be to flip the bit value.

Given an initial configuration A, a new configuration A^* is obtained as explained above by flipping a bit. Then, the difference of the likelihood of the new configuration and the old configuration is calculated by

$$\Delta U = U(A^*) - U(A) = \sum_{a^* \in \mathcal{C}_1} V_1(a^*) + \sum_{a^*, b^* \in \mathcal{C}_2} V_2(a^*, b^*) + \sum_{a^*, b^*, c^* \in \mathcal{C}_3} V_3(a^*, b^*, c^*) + \dots$$
$$- \sum_{a \in \mathcal{C}_1} V_1(a) + \sum_{a, b \in \mathcal{C}_2} V_2(a, b) + \sum_{a, b, c \in \mathcal{C}_3} V_3(a, b, c) + \dots$$

and the new configuration is accepted with probability 1 if the new likelihood is higher than the old one. Otherwise, the new configuration is accepted with probability $e^{-\Delta U/T}$, so the probability of accepting the new configuration becomes:

$$P(A \to A^*) = \begin{cases} 1 & \Delta U \geq 0 \\ e^{-\Delta U/T} & \Delta U < 0 \end{cases}$$

A pass through all the components in A in this way is called a *sweep* over the variables in A. In our case, we generate a sample from the distribution by doing 4 sweeps over A, resulting in the evaluation of a total of $4n$ new configurations. This should result in a sample that has high enough probability to be detected by the estimation algorithm described above.

5. Application to the two-dimensional binary asymmetric channel

5.1 Channel model

In this section, we will apply this general method for cluster error detection and correction to the binary asymmetric channel As the name of the channel implies, we will assume that errors are asymmetric so that only the transition $-1 \to 1$ occurs in a received codeword.

Definition 2 (Matrix OR). *Assume A and B are matrices with dimensions $d_1 \times d_2$ where $d_1 \cdot d_2 = n$, and with coordinates a_i and b_i respectively. Then the OR of these matrices is defined as*

$$A \vee B \triangleq a_i \vee b_i, \qquad 1 \leq i \leq n$$

The received word Y can then be defined as the combination of

$$Y = C \vee X$$

Y - received word
C - original codeword
X - error pattern
$\vee$ - OR operator on matrices as defined above

5.2 System model

Information I is encoded, producing a codeword C. For each generated codeword C, the channel induces two-dimensionally correlated noise X and the resulting word Y is passed to the decoder. Information is converted to likelihood ratios and sent to the SISO. The output

likelihood values L from the SISO are then sent to the detector. Based on these values, the detector produces an estimate of the error pattern that was multiplied with the codeword. This information is subsequently fed back into the SISO in the next iteration. The decoding process continues until either the SISO finds a valid codeword, or a set of stopping criteria is reached. The final estimate of the codeword is produced by the SISO. See figure 2.

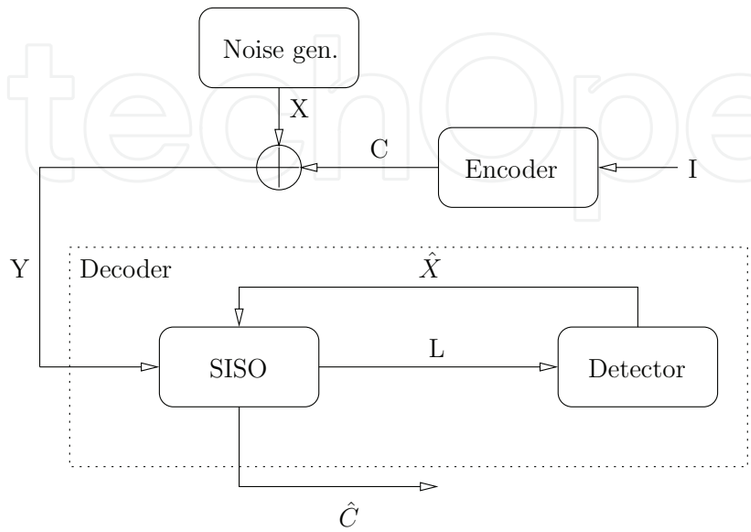


Fig. 2. System model

5.3 MAP estimation

To optimize the maximum a priori probability, we need to find some expression for the conditional probability $P(Y = y|X = x)$ as noted in chapter 4.4. Based on the assumptions in chapter 4.4, the conditional probabilities $P(Y_i = y_i|X_i = x_i)$ for each information symbol is given in Table 1.

x \ y	1	−1
1	$\frac{1}{2}$	$\frac{1}{2}$
−1	0	1

Table 1. Transition probabilities

The conditional probabilities in the table can be expressed as an exponential function by

$$P(Y_i = y_i|X_i = x_i) = \lim_{\epsilon \to 0} \frac{1}{2} \exp\left[\frac{-y_i(1 - x_i)\ln 2}{1 - y_i + \epsilon}\right]$$

We can then express the probability of a given y conditioned on a configuration x by

$$P(Y = y|X = x) = \prod_i \lim_{\epsilon \to 0} \frac{1}{2} \exp\left[\frac{-y_i(1 - x_i)\ln 2}{1 - y_i + \epsilon}\right] \tag{5}$$

Substituting (1) and (5) into (4), we can find the joint probability by

$$P(X=x,Y=y)=\left[Z^{-1}e^{-\frac{1}{T}U(x)}\right]\prod_i \lim_{\epsilon\to 0}\frac{1}{2}\exp\left[\frac{-y_i(1-x_i)\ln 2}{1-y_i+\epsilon}\right] \tag{6}$$

Since the natural logarithm is strictly increasing, the following equality holds:

$$\arg\max_X(P(X,Y))=\arg\max_X(\ln(P(X,Y))) \tag{7}$$

In order to avoid computing Z in (6), we take the logarithm of both sides and eliminate constants to get

$$V(x)=U(x)+\sum_i \lim_{\epsilon\to 0}\left[\frac{-y_i(1-x_i)\ln 2}{1-y_i+\epsilon}\right]$$

where $V(x)=\ln\left[P(X=x,Y=y)\right]$.

We define the partial functions V_i of $U(x)$ according to Li (2000); Shridhar et al. (1989)

$$V_1(x_i) = \alpha x_i$$

$$V_2(x_i,x_{i'}) = \beta_{i,i'}x_i x_{i'}$$

$$V_3(x_i,x_{i'},x_{i''}) = \cdots = 0$$

Note that the expression for V_2 implies that $V_2(x_i,x_{i'})=\beta_{i,i'}$ for $x_i = x_{i'}$and $V_2(x_i,x_{i'}) = -\beta_{i,i'}$for $x_i \neq x_{i'}$.

From this we get a new expression for $V(x)$:

$$V(x)=\sum_i\left[\alpha x_i+\beta_{i,i'}\sum_{i'\in\mathcal{N}_i}x_i x_{i'}\ +\lim_{\epsilon\to 0}\left[\frac{-y_i(1-x_i)\ln 2}{1-y_i+\epsilon}\right]\right],$$

and splitting the last term into a constant and a non-constant term yields

$$V(x)=\sum_i\left[\alpha x_i+\beta_{i,i'}\sum_{i'\in\mathcal{N}_i}x_i x_{i'}\ +\lim_{\epsilon\to 0}\left[\frac{x_i y_i\ln 2}{1-y_i+\epsilon}\right]+\lim_{\epsilon\to 0}\left[\frac{-y_i\ln 2}{1-y_i+\epsilon}\right]\right].$$

Since the last term only depends on y, we can find the MAP configuration by simplifying the expression to:

$$\begin{aligned}\mathcal{V}(x)&=\sum_i\left[\alpha x_i+\beta\sum_{i'\in\mathcal{N}_i}x_i x_{i'}+\lim_{\epsilon\to 0}\left[\frac{x_i y_i\ln 2}{1-y_i+\epsilon}\right]\right]\\&=\sum_i\left[\alpha+\beta\sum_{i'\in\mathcal{N}_i}x_{i'}+\lim_{\epsilon\to 0}\left[\frac{y_i\ln 2}{1-y_i+\epsilon}\right]\right]x_i\end{aligned} \tag{8}$$

Having obtained an estimate

$$\hat{X}=\{\hat{X}_1,\ldots,\hat{X}_i,\ldots,\hat{X}_n\}$$

of the error pattern, it can be used to find likelihood ratios for input to the decoder. For each bit, we set the likelihood ratio to

$$L_i = \frac{P(C_i = -1|Y_i, \hat{X}_i)}{P(C_i = 1|Y_i, \hat{X}_i)}$$

The resulting probabilities can be seen in Table 2. In the table, ρ is the probability that a bit

$\hat{X}$ \ Y	-1	1
-1	$\frac{1-\rho}{\rho}$	$\frac{\rho}{1-\rho}$
1	$\frac{\rho}{1-\rho}$	$\frac{1-\rho}{\rho}$

Table 2. Input probabilities to the decoder

belonging to the error pattern is incorrectly estimated as a 1-bit. The parameter ρ must be estimated by simulation, but should in general be small, indicating a relatively certain -1-bit.

5.4 Performance of estimation algorithm

The performance of the estimation algorithm depends heavily on the value of β, which determines the degree of clustering in the error pattern. A critical performance parameter is the probability ε that not all bits in the error pattern are detected by the estimation algorithm. A bit that belongs to the error pattern, but is not detected as such, is given a high probability of being correct, and can hence be the source of errors that are hard to correct. Therefore, ε is an important measure of the reliability of the algorithm. As can be seen in Fig. 3, $P(\varepsilon)$ is high for $\beta < 0.5$, reflecting the fact that small and very irregular error clusters appear in this range. $P(\varepsilon)$ drops sharply initially, but levels out when $\beta > 1$ as a result of the clusters becoming bigger and more coherent. As we shall see later, this is also reflected in the performance of the algorithm.

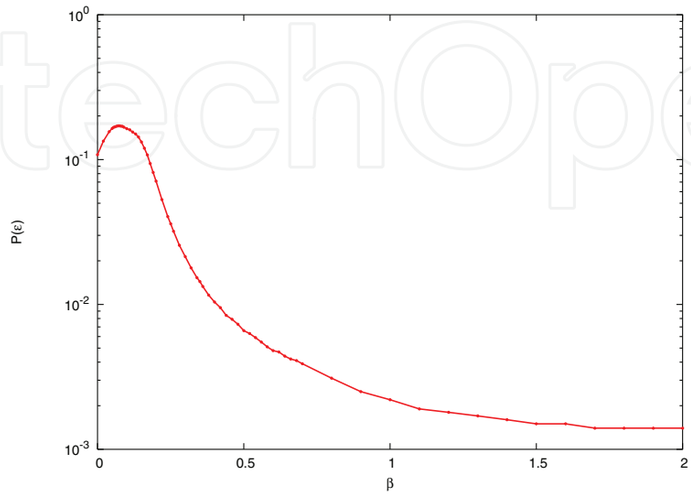


Fig. 3. $P(\varepsilon)$ for fixed $\beta = 0.2$ in the estimation algorithm.

5.5 Results

A regular LDPC code was used as the error correcting code component , with different values of β in the simulations, and $\alpha = \gamma = \cdots = 0$ in the estimation algorithm. We assumed that the receiver does not know the value of β used by the noise generating process. The components of the simulator was then connected as shown in Fig. 6 The simulations show that there is a large performance gain for some choices of parameter using the LDPC-MRF combination described above. The value of β has great influence over the relative performance of the two decoding methods. Looking in Fig. 5 at the performance of a code in combination with the MAP error estimate and alone, under varying β, we can observe that the performance difference between the two decoders increases as β increases. This is due to the effect described in Section 5.4: as β increases, the reliability of the error estimate also increases. We also notice that the drop in BER levels off at about $\beta = 1$ corresponding to the reliability of the estimate leveling off from the same point. The performance of the decoder could also be measured under varying bit error probabilities, but because the bit error probability depends

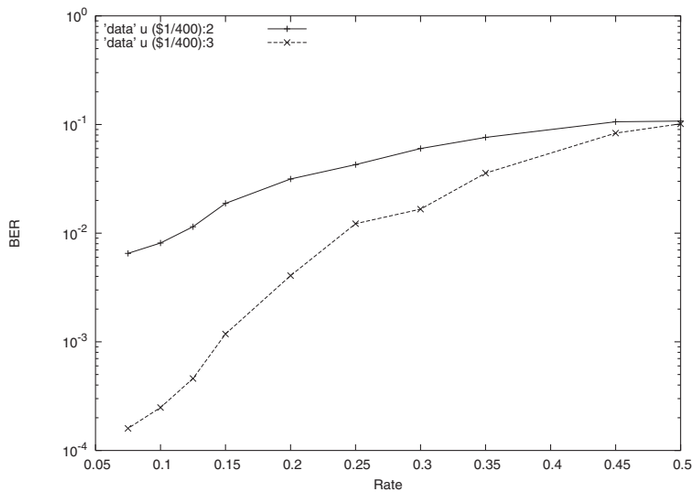


Fig. 4. Performance under varying rate with $\beta = 0.2$ and $\beta = 1.0$

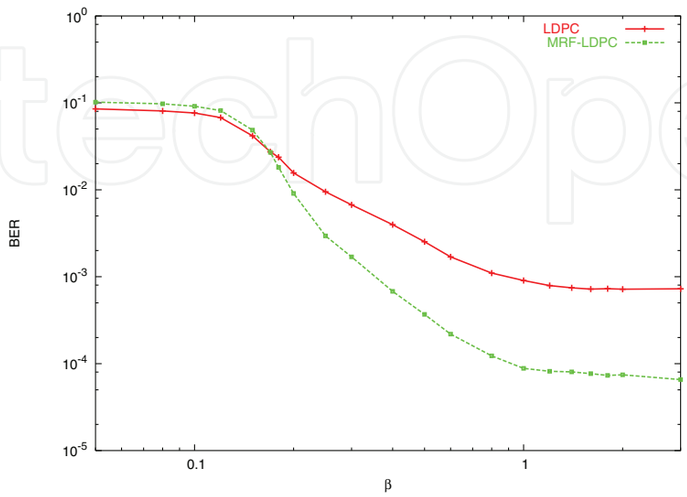


Fig. 5. Performance under varying β with rate $\frac{3}{8}$

on the parameter β in the Gibbs distribution in a way that makes it hard to predict the average error probability over codewords, we fix the value of β to $\beta = 0.2$ and $\beta = 1.0$ which gives an average error rate of about 0.12 and 0.02 respectively, and study the performance of joint LDPC - MRF decoding for different code rates using these parameters. We see in Fig. 4 that the effect of the MRF estimator gives very good results in combination with the LDPC code when the code rate is sufficiently low, while the performance gap between the two decoders gets smaller as the code rate grows. This occurs because the MRF-LDPC decoder needs a certain amount of information from the code itself to determine the value of the bits in the error pattern, even if the MRF estimator provides a perfect estimate of the errors.

6. Application to the two-dimensional binary symmetric channel

6.1 Channel model

In this section we will see how the outlined cluster detection technique can be applied to the two-dimensional binary symmetric channel. On this channel, both the transition $1 \to -1$ and $-1 \to 1$ may take place.

Definition 3 (Componentwise product)**.** *Assume A and B are matrices with the same dimensions $d_1 \times d_2$ where $d_1 \cdot d_2 = n$, and with coordinates a_i and b_i respectively. Then the componentwise product of these matrices is defined as*

$$A * B = a_i \cdot b_i, \qquad 1 \leq i \leq n$$

Now assuming X, Y and C are $d_1 \times d_2$ matrices with bipolar coordinates, the effect of the channel can be described as:

$$Y = C * X$$

where
Y - received word
C - original codeword
X - error pattern
$*$ - multiplicative operator on matrices defined as above

6.2 System model

Like in chapter 5.2, information I is encoded, producing a codeword C. Two-dimensionally correlated noise X is applied to the codeword and the result Y is passed to the decoder. The decoding process is different in this case, however. The likelihood values L from the SISO are used to find some values δ that measures the distance between the received input and the output of the SISO, multiplied by the channel value. The δ value can be seen as an estimate of the value of the corresponding bit in X. These values are sent to the cluster detector to be used as basis for producing an estimate of the error cluster which is in turn fed back to the SISO. As in 5.2, the iterative process continues until a valid codeword is found or a set of stopping criteria is met.

6.3 MAP estimation

6.3.1 distribution of Δ_i

The values Y received from the channel is used to compute likelihood ratios $L^I = \{L_1^I, L_2^I, \ldots, L_n^I\}$ for the bits. Since we assume that the variables are bipolar, the likelihood

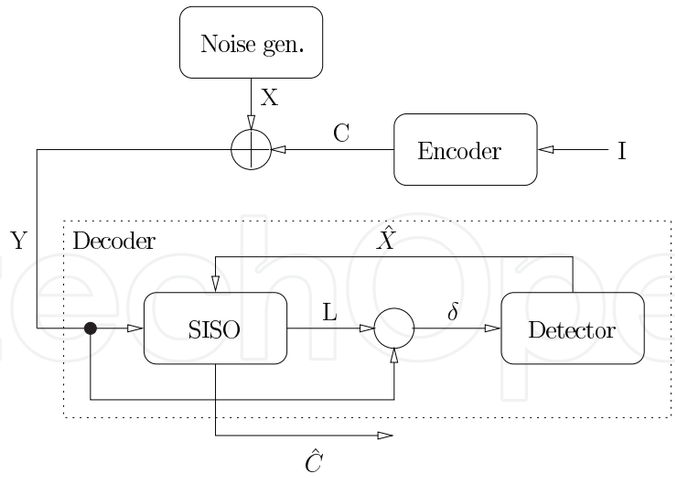


Fig. 6. System model

ratios can be expressed as

$$L_i^I = \frac{P(C_i = -1|Y_i)}{P(C_i = +1|Y_i)}.$$

These values are the channel values used in the SISO. The soft output from the SISO-component in the decoder is $L^O = \{L_1^O, L_2^O, \ldots, L_n^O\}$. The output L^O of the SISO also has the form of likelihood ratios. We now take the logarithm of L^I and L^O giving us the values $\widetilde{L}^I$ and $\widetilde{L}^O$. These variables are now real valued in the range $\langle -\infty, \infty \rangle$, with a negative value indicating a possible -1 bit and a positive value indicating a possible $+1$ bit. The difference $\widetilde{L}_i^I - \widetilde{L}_i^O$ measures the distance between the input- and output values, and multiplication by Y_i gives the relative direction Δ_i of the change.

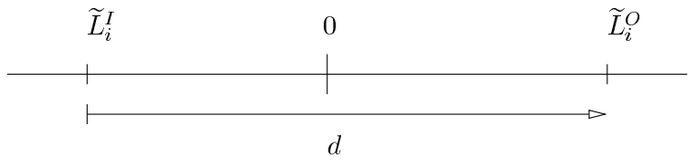


Fig. 7. Negative Δ_i

As an example, in figure 7 the distance between the input and the output d is positive, but $Y_i = -1$, so the relative direction $\Delta_i = -1 \cdot d = -d$ is negative. This corresponds to a higher probability that the bit was flipped by the channel.

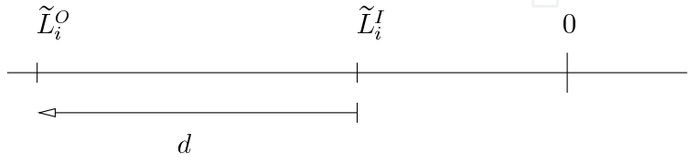


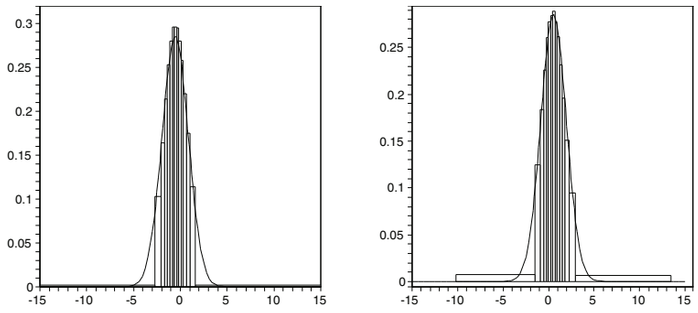
Fig. 8. Positive Δ_i

In figure 8 on the other hand, the distance d is negative and $Y_i = -1$ so the relative distance $\Delta_i = -d$ is positive, indicating a higher probability that the bit is correct.

The values $\Delta_i = \Upsilon_i(\widetilde{L}_i^I - \widetilde{L}_i^O)$ are sent to the detector, which computes the MAP estimate of the configuration of X, i.e. the most likely X to produce the observed values. As is shown below, knowing the distribution of $P(\Delta_i = \delta_i \mid X_i = x_i)$ is essential to the MAP estimate calculation, so we shall make the assumption that the information from the SISO, Δ_i, conditioned on X_i has a normal distribution with variance σ and mean $X_i\mu$. The conditional probabilities can then be expressed as an exponential function by approximating them with the normal distribution with mean $x_i\mu$ so that

$$P(\Delta_i = \delta_i \mid X_i = x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\delta_i - x_i\mu)^2}{2\sigma^2}}. \tag{9}$$

As an example of this, we can see in Fig. 9(a) and Fig. 9(b) the distribution of the Δ_i's for $X_i = -1$ and $X_i = +1$ respectively, compared with the normal distribution with $\mu = \pm 0.55$ and $\sigma = 1.4$. We can see that this approximation is reasonably good. MAP estimation of the



(a) $X_i = +1$ and $\mu = 0.55$ (b) $X_i = -1$ and $\mu = -0.55$

Fig. 9. Distribution of the Δ_i's

error pattern X based on the values Δ from the SISO, can be formulated as the maximization of the *a posteriori* probability $P(X = x|\Delta = \delta)$ with respect to x. That is, we want to find a configuration x that makes the probability $P(X = x|\Delta = \delta)$ as high as possible for a given δ. Bayes rule gives us

$$P(X = x|\Delta = \delta) = \frac{P(X = x)P(\Delta = \delta|X = x)}{P(\Delta = \delta)} \tag{10}$$

Since we are optimizing the expression for a given value of δ, we can maximize over

$$P(X = x)P(\Delta = \delta|X = x) \tag{11}$$

instead of (10).

Based on the assumption that the distribution of $P(\Delta_i = \delta_i \mid X_i = x_i)$ is given by (9), we can express the probability of a given δ conditioned on a configuration x by

$$P(\Delta = \delta \mid X = x) = \prod_i P(\delta_i \mid x_i) = \prod_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\delta_i - x_i\mu)^2}{2\sigma^2}} \tag{12}$$

Substituting (1) and (12) into (11), and taking $T = 1$, we can express the product of the probabilities as:

$$P(X = x)P(\Delta = \delta \mid X = x) = \left[Z^{-1}e^{-U(x)}\right] \prod_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\delta_i - x_i\mu)^2}{2\sigma^2}}$$

In order to avoid computing Z in the above expression, we take the logarithm of both sides and eliminate the constants to get

$$V(x) = U(x) - \sum_i \frac{(\delta_i - x_i\mu)^2}{2\sigma^2}. \tag{13}$$

Since the natural logarithm is strictly increasing, optimizing (13) with respect to x also optimizes (5).

We define the partial functions V_t of $U(x)$ in (3) according to Li (2000) and Shridhar et al. (1989):

$$\sum_{c\in\mathcal{C}_1} V_1(c) = \sum_i \alpha x_i$$

$$\sum_{c\in\mathcal{C}_2} V_2(c) = \sum_i \sum_{x_{i'}\in\mathcal{N}_i} \beta x_i x_{i'} \quad i \neq i'$$

$$V_t(\cdot) = 0 \quad \forall t > 2$$

This implies that we let the total probability depend on the value of each bit represented by αx_i and the value of each neighboring bit represented by $\beta x_i x_{i'}$, while we do not consider more complex dependencies like three-ways dependencies and up. Note that the expression for V_2 implies that $V_2(x_i, x_{i'}) = \beta$ for $x_i = x_{i'}$and $V_2(x_i, x_{i'}) = -\beta$ for $x_i \neq x_{i'}$.

From this we get a new expression for $V(x)$:

$$V(x) = \sum_i \left[\alpha x_i + \beta \sum_{x_{i'}\in\mathcal{N}_i} x_i x_{i'} - \frac{(\delta_i - x_i\mu)^2}{2\sigma^2} \right],$$

and expanding the last term of the sum yields

$$V(x) = \sum_i \left[\alpha x_i + \beta \sum_{x_{i'}\in\mathcal{N}_i} x_i x_{i'} - \frac{(\delta_i^2 - 2x_i\delta_i\mu + x_i^2\mu^2)}{2\sigma^2} \right].$$

The variables x_i are bipolar so $x_i^2 = 1$ and we can simplify the expression to:

$$\begin{aligned} \mathcal{V}(x) &= \sum_i \left[\alpha x_i + \beta \sum_{x_{i'}\in\mathcal{N}_i} x_i x_{i'} - \frac{(\delta_i^2 - 2x_i\delta_i\mu + \mu^2)}{2\sigma^2} \right] \\ &= \sum_i \left[\alpha x_i + \beta \sum_{x_{i'}\in\mathcal{N}_i} x_i x_{i'} - \frac{\delta_i^2}{2\sigma^2} + \frac{2x_i\delta_i\mu}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} \right] \end{aligned}$$

As we are doing a maximization with respect to x_i, any term in the sum that does not contain or otherwise depend on x_i will not affect the result, and hence we cancel the terms $-\frac{\delta_i^2}{2\sigma^2}$ and

$-\frac{\mu^2}{2\sigma^2}$, and the expression to maximize over becomes:

$$\begin{aligned}\mathcal{V}(x) &= \sum_i \left[\alpha x_i + \beta \sum_{x_{i'} \in \mathcal{N}_i} x_i x_{i'} + \frac{2x_i \delta_i \mu}{2\sigma^2} \right] \\ &= \sum_i \left[\alpha + \beta \sum_{x_{i'} \in \mathcal{N}_i} x_{i'} + \frac{\mu}{\sigma^2} \delta_i \right] x_i \end{aligned} \tag{14}$$

6.3.2 Optimization of $\mathcal{V}(x)$

To do a global optimization of the expression above with respect to x would become computationally infeasible as the size of x increases. Instead we can use the local dependencies between bits to do a local optimization along the lines of the PDFE in Neifeld & King (1998); Neifield et al. (1996) or the partial binary segmentation algorithm of Shridhar et al. (1989). It is apparent that when $\mathcal{V}(x)$ is expressed as in (14), we can always choose the value of x_i so that each term in the sum becomes positive, and thus the sum is non-decreasing. For each node we compute the value of

$$\alpha x_i + \beta \sum_{x_{i'} \in \mathcal{N}_i} x_{i'} + \frac{\delta_i \mu}{\sigma^2}$$

and set the value of x_i so that the product is positive. This procedure is iterated until we converge on a solution where all terms in the sum are positive, or a maximum number of iterations is reached. Normally the process arrives at a solution after less than 10 iterations.

Having obtained an estimate

$$\hat{X} = \{\hat{X}_1, \ldots, \hat{X}_i, \ldots, \hat{X}_n\}$$

of the error pattern, it can be used to find likelihood ratios for input to the SISO. For each bit, we set the likelihood ratio to

$$L_i = \frac{P(C_i = -1 | Y_i, \hat{X}_i)}{P(C_i = 1 | Y_i, \hat{X}_i)}$$

The resulting probabilities can be seen in Fig. 10. In the table, ρ is the probability that a bit

$\hat{X}$ \ Y	-1	1
-1	$\frac{1-\rho}{\rho}$	$\frac{\rho}{1-\rho}$
1	$\frac{\rho}{1-\rho}$	$\frac{1-\rho}{\rho}$

Fig. 10. Input probabilities to the decoder

detected as belonging to the error pattern is actually a -1-bit, i.e. $\rho = P(X_i = -1 \mid \hat{X}_i = -1)$. For a given channel, the parameter ρ must be found experimentally by sending some known codewords over the channel. When the receiver knows the value of X, the value of ρ can be computed based on the value of the estimates $\hat{X}$. ρ should in general be large indicating a relatively certain -1-bit.

6.4 Results

We have implemented the system with an LDPC decoder as the SISO component, and we assume a priori knowledge of the values α, β and ρ. The error-simulation process allows upper bounding of the overall error rate of the channel, and the performance of the decoder is measured for two different upper bounds on the error rate, denoted E_h, to investigate the effect of varying error rates on the performance of the joint decoding and estimation. The parameter β in the Gibbs distribution should ideally depend on the the bit error probability of the channel, but in the simulations we have chosen to fix the value of β to $\beta = 0.2$ which corresponds to an average error rate of about 0.12, and study the performance of joint LDPC - MRF decoding for different code rates under this assumption. This is a rather "harsh" assumption in the sense that in practice it should be possible to find an estimate of the value of β before transmission takes place. The performance of the joint decoding and estimation algorithm is compared to decoding of the same received information using the same LDPC decoder component but under a random error assumption, i.e., the decoder assumes that there are no dependencies in the error generating process. The LDPC codes used in the simulations are generic regular LDPC codes that were not optimized for use on this particluar type of channel. We see in Fig. 11 that the effect of the MRF estimator gives very good results in combination with the LDPC code when the code rate is sufficiently low, while the performance gap between the two decoders gets smaller as the code rate grows. This indicates that even if the MRF estimator provides a perfect estimate of the errors, the MRF-LDPC decoder still needs a certain amount of information from the code itself to determine the value of the bits in the error pattern, and therefore the mutual gain when exchanging information between the LDPC component and the channel detector component decreases as the code rate increases.

Fig. 12 shows that the performance of an LDPC-decoder in combination with the MAP error estimate, relative to an LDPC code alone. We see that the gain is even greater for a higher upper bound on the error rate, mainly because the MRF-estimate makes the most difference in face of a high number of errors.

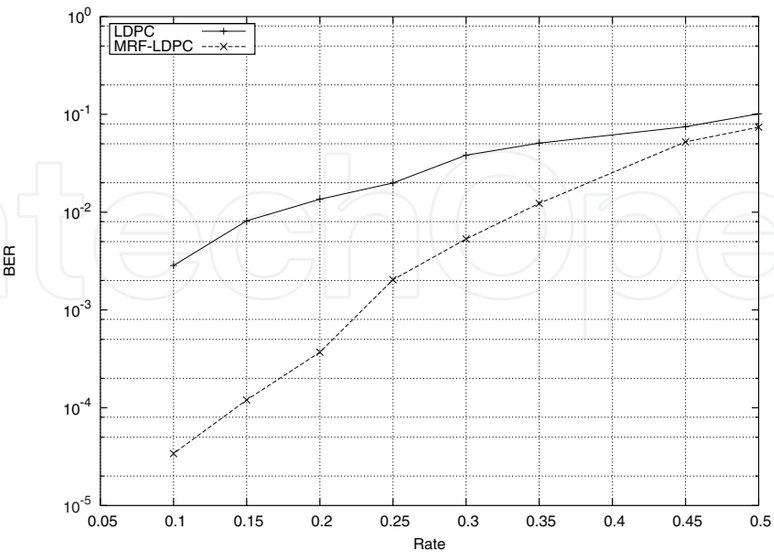


Fig. 11. Performance under varying rate with $E_h = 0.1250$

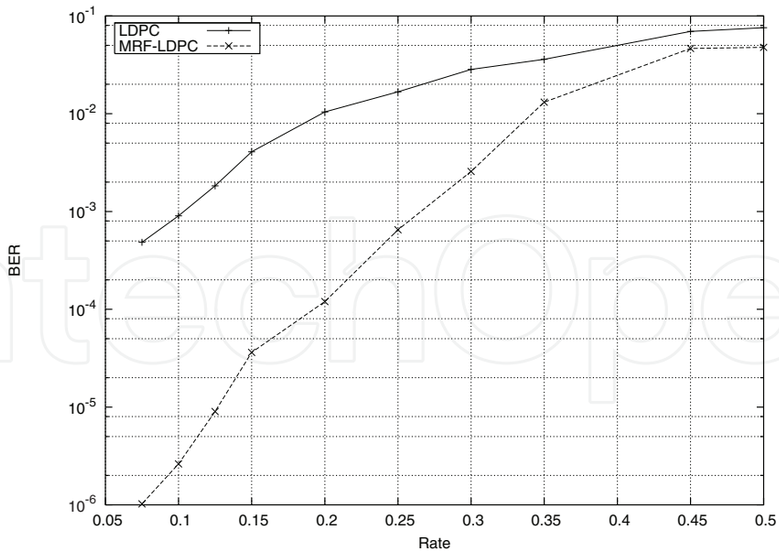


Fig. 12. Performance under varying rate with $E_h = 0.2250$

7. Conclusion

By applying principles known form digital image restoration, we have introduced a channel model for two-dimensional channels with memory based on Markov random fields which allows us to describe spatially dependent errors. We have showed that a significant performance gain over an ordinary error correcting code can be achieved , for both the symmetric and the asymmetric binary two-dimensional channel by combining an error-correcting component with an MRF-based burst detection algorithm. We have also demonstrated that this decoding technique gives the most gain for larger clusters and for lower information rates.

8. References

Banham, M. & Katsaggelos, A. (1997). Digital image restoration, *Signal Processing Magazine, IEEE* 14(2): 24–41.

Blaum, M., Bruck, J. & Vardy, A. (1998). Interleaving schemes for multidimensional cluster errors, *Information Theory, IEEE Transactions on* 44(2): 730–743.

Breitbach, M., Bossert, M., Zyablov, V. & Sidorenko, V. (1998). Array codes correcting a two-dimensional cluster of errors, *Information Theory, IEEE Transactions on* 44(5): 2025–2031.

Chalmond, B. (1988). Image restoration using an estimated markov model, *Signal Processing* 15(2): 115–129.

El-Gabali, M., Shridhar, M. & Ahmadi, M. (1987). Segmentation of noisy images modelled by markov random fields with gibbs distribution, *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, Vol. 12, IEEE, pp. 551–554.

El-Gabali, M., Shridhar, M. & Ahmadi, M. (1988). Restoration of degraded images using markov random fields, *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, IEEE, pp. 1004–1007.

El-Gabali, M., Shridhar, M., Ahmadi, M. & Kekre, J. (n.d.). Restoration of blurred images, *Signals, Systems and Computers, 1990. 1990 Conference Record Twenty-Fourth Asilomar Conference on*, Vol. 2, IEEE, p. 549.

Ellingsen, P. & Kvamme, A. (2010). Iterative decoding and estimation of two-dimensional channels with memory, *2010 Third International Conference on Communication Theory, Reliability, and Quality of Service*, IEEE, pp. 43–48.

Ellingsen, P., Ytrehus, O. & Siegel, P. (n.d.). Enhanced decoding by error detection on a channel with correlated two-dimensional errors, *Information Theory Workshop, 2004. IEEE*, IEEE, pp. 17–21.

Farrell, P. (1982). Array codes for correcting cluster-error patterns, *Proc. IEE Conf. Elect. Signal Processing (York, England)*.

Geman, S. & Geman, D. (1993). Stochastic relaxation, gibbs distributions and the bayesian restoration of images, *Journal of Applied Statistics* 20(5-6): 25–62.

Gilbert, E. et al. (1960). Capacity of a burst-noise channel, *Bell Syst. Tech. J* 39(9): 1253–1265.

Jeng, F. & Woods, J. (1991). Compound gauss-markov random fields for image estimation, *Signal Processing, IEEE Transactions on* 39(3): 683–697.

Kindermann, R. & Snell, J. L. (1980). Markov random fields and their applications.

Kurkoski, B. M. (2008). Towards efficient detection of two-dimensional intersymbol interference channels.

Li, S. Z. (2000). Modeling image analysis problems using markov random fields, Vol. 20, pp. 1–43.

Neifeld, M. . & King, B. M. (1998). Parallel detection algorithms for page oriented optical memories. *Applied optics*(37(26) pp. 6275–6298

Neifield, M. A., Chugg, K. M. & King, B. M. (1996). Parallel data detection in page-oriented optical memory, *Optics Letters* 21(18): 1481–1483.

Schwartz, M. & Etzion, T. (2003). Optimal 2-dimensional 3-dispersion lattices, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes* pp. 606–606.

Schwartz, M. & Etzion, T. (2005). Two-dimensional cluster-correcting codes, *IEEE Trans. Info. Theory* 51: 2121–2132.

Shridhar, M., Ahmadi, M. & El-Gabali, M. (1989). Restoration of noisy images modeled by markov random fields with gibbs distribution, *Circuits and Systems, IEEE Transactions on* 36(6): 884–890.

Singla, N. & O'Sullivan, J. A. (2005). Joint equalization and decoding for nonlinear two-dimensional intersymbol interference channels.

Xu, W. & Golomb, S. (2007). Optimal interleaving schemes for correcting two-dimensional cluster errors, *Discrete applied mathematics* 155(10): 1200–1212.

Zhang, J. (1993). The mean field theory in em procedures for blind markov random field image restoration, *Image Processing, IEEE Transactions on* 2(1): 27–40.

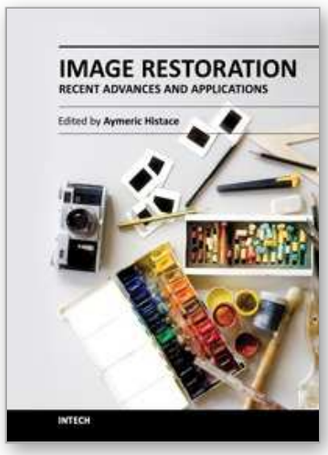


Image Restoration - Recent Advances and Applications

Edited by Dr Aymeric Histace

ISBN 978-953-51-0388-2
Hard cover, 372 pages
Publisher InTech
Published online 04, April, 2012
Published in print edition April, 2012

This book represents a sample of recent contributions of researchers all around the world in the field of image restoration. The book consists of 15 chapters organized in three main sections (Theory, Applications, Interdisciplinarity). Topics cover some different aspects of the theory of image restoration, but this book is also an occasion to highlight some new topics of research related to the emergence of some original imaging devices. From this arise some real challenging problems related to image reconstruction/restoration that open the way to some new fundamental scientific questions closely related with the world we interact with.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen