Statistical Mechanical Models for Image Processing

Vom Fachbereich Physik der
Carl von Ossietzky Universität Oldenburg
zur Erlangung des Grades eines

Doktors der Naturwissenschaften

angenommene Dissertation.

Thorsten Wanschura
geb. am 13. Januar 1971
in Oldenburg

Erstgutachter
Prof. Dr. Pál Rufján
Zweitgutachter
Prof. Dr. Martin Holthaus

Datum der Disputation: 16. Oktober 2001

Statistical Mechanical Models for Image Processing

Vom Fachbereich Physik der
Carl von Ossietzky Universität Oldenburg
zur Erlangung des Grades eines

Doktors der Naturwissenschaften

angenommene Dissertation.

Thorsten Wanschura
geb. am 13. Januar 1971
in Oldenburg

Erstgutachter
Prof. Dr. Pál Ruján
Zweitgutachter
Prof. Dr. Martin Holthaus

Datum der Disputation: 16. Oktober 2001

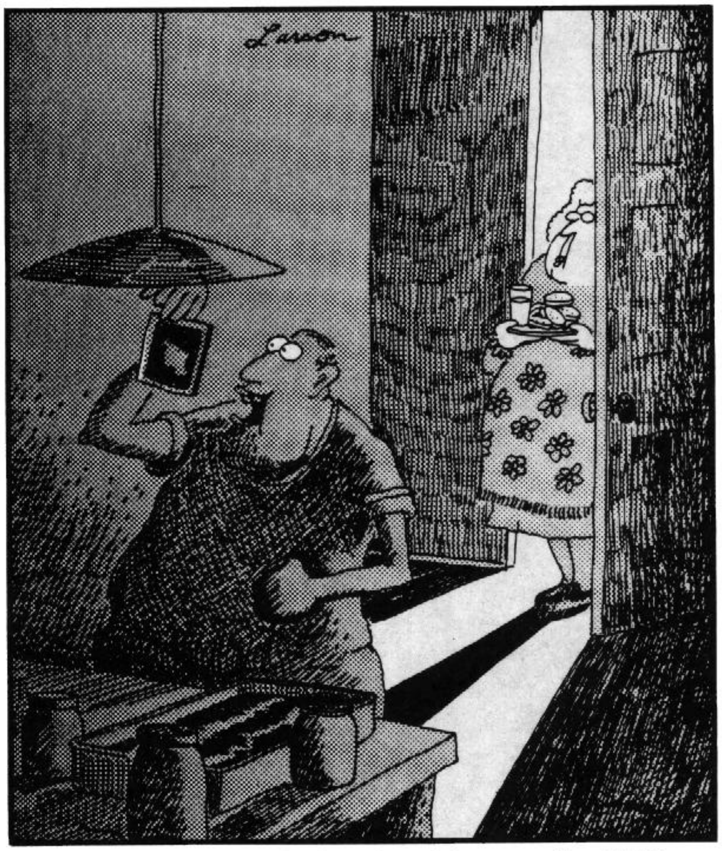
Preface

In 1994, during a year at the University of Exeter (UK), I was given the chance to work with MRI images, and it was then that I became interested in image processing, understanding that this field contains a multitude of exciting applications both from the practical and theoretical point of view. Luckily, a group involved with the SFB 517 Neurocognition in Oldenburg was investigating models of the visual system with a special interest in the retina. Looking at it from yet another aspect, this time motivated by biology and statistical physics, I started the work which is presented in the following chapters.

At this point, I would like to thank my mentor Prof. Dr. Pál Ruján, as well as the other (former) members of the (former) AG spÎn at the Carl von Ossietzky Universität Oldenburg, Dr. Harry Urbschat and Johannes Hausmann. A special thank goes to Prof. Dr. Ulrich Ramacher at Infineon Technologies AG who organized the financial support for this thesis in a very unbureaucratic way.

Finally, to motivate and convince a potential reader of this work with the striking blow of reality, the next page illustrates the need for image processing. In this case however, we might be too late ?

This work was supported by Infineon Technologies AG.



"I've done it! The first real evidence of a UFO! ...
And with my own camera, in my own
darkroom, and in my own ..."

Contents

1 Introduction **1**

2 Information Theory and Learning **5**

- 2.1 Shannon Model 5
  - 2.1.1 Self-information 6
  - 2.1.2 Entropy 7
  - 2.1.3 Coding and source coding 7
  - 2.1.4 Channel coding and capacity 8
- 2.2 Models provide information 8
  - 2.2.1 Independent data 9
  - 2.2.2 Markov chains 9
  - 2.2.3 Statistical physics 10
- 2.3 Learning and memory 10
- 2.4 Bayesian methods 11

3 Image and error statistics **13**

- 3.1 Natural images 13
- 3.2 Image statistics 14
  - 3.2.1 Zero-order statistics 14
  - 3.2.2 First order statistics – Markov model 17
  - 3.2.3 A quick glance at the Zq-model 19
  - 3.2.4 Entropy 19
- 3.3 Common image models 21
- 3.4 Noise 22
  - 3.4.1 Stochastic noise 22
  - 3.4.2 Correlated noise – filters and blurring 25
  - 3.4.3 Estimating noise from the data 27
- 3.5 Image quality 28

4 Markov Random Fields 29
4.1 Introduction 29
4.2 Markov Random Fields and Gibbs Random Fields 30
4.2.1 Definition of a Markov Random Field 31
4.2.2 Definition of the Gibbs Random Field 32
4.2.3 Equivalence of Markov and Gibbs Random Fields . . . 32
4.3 Markov Random Fields and image models 33
4.3.1 Modeling an image 33
4.4 Restoring images 35
4.4.1 Maximum a posterior (MAP) 36
4.4.2 Marginal a posteriori mode (MPM) 36
4.4.3 Thresholded posterior mean (TPM) 36
4.4.4 Coming to Monte Carlo 37

5 New physical models for images 39
5.1 Physical background 39
5.1.1 Ising Model – a one-dimensional information system . 40
5.2 Finding the coupling constants of an image 43
5.3 Two-dimensional models 43
5.3.1 Solution to the transfer operator eigenvalue problem . 45
5.3.2 Two-dimensional Ising model for black and white images 47
5.3.3 Two-dimensional Z_q model for grey level images . . . 49
5.4 One-dimensional models 52
5.4.1 One-dimensional Ising model for black and white images 52
5.4.2 One-dimensional Z_q model for grey level images 54
5.4.3 Ashkin-Teller model for grey level images 58
5.4.4 One-dimensional Z_q model for colour images 62
5.5 Restoration of Images 63
5.5.1 Bayes restoration 63
5.5.2 Metropolis-Rosenbluth-Rosenbluth-Teller-Teller algorithm 64
5.5.3 Optimal decoding 65
5.5.4 Rank-Order Monte Carlo 66
5.6 Classification – Texture Segmentation with the Z_q model: a new approach 66
5.7 Noise estimation and model verification 67

6 Experiments and Applications 70
6.1 Two dimensional Ising model 70
6.2 One dimensional Z_q model 74

6.2.1 Correlations and random noise 74
6.2.2 Rank-Order scheme 74
6.2.3 Close to the prior information 78
6.2.4 Prior-only Monte Carlo 80
6.2.5 Random noise – high noise regime 82
6.2.6 Learning from the noisy observation 82
6.2.7 White noise 85
6.2.8 Masking and the human visual system (HVS) 87
6.2.9 Deblurring 89
6.2.10 Colour images 89
6.3 Noise estimation 90
6.4 Texture segmentation 92

7 Feature extraction using Transformed Probabilities 96
7.1 Introduction and Motivation 96
7.2 The problem stated 98
7.3 Independent Component Analysis 98
7.4 Estimation of the probability density function 100
7.5 Labeling the data 101
7.6 The whole program 102
7.7 Applications 102
7.7.1 Phantom test image 104
7.7.2 Isolating the torn edge of a paper 104

8 Conclusion 109

Abstract

This thesis introduces a solution to the problem of image restoration and feature extraction by incorporating new image models derived from statistical physics. A special lattice spin Hamiltonian is used which is well suited for both source coding and for modeling information loss within the Bayesian framework. The parameter estimation problem is solved analytically using transfer-matrix methods.

Beyond its inherent practical usefulness the image restoration problem illustrates directly basic concepts related to information theory, statistical inference, and perception.

The work is split in two parts: Chapters 1 to 4 contain a summary of the problem and existing models, Chapters 5 to 7 introduce the new models and illustrate their capabilities in a variety of experiments. The conclusion can be found in Chapter 8.

Chapter 1

Introduction

With the advent of modern information processing systems (such as computers) which are capable of handling a vast amount of data, one special interest – combined with a certain sense of fascination – emerged: the processing of images. Although this might first look like a rather technical problem it is in fact an area which occupies a large number of disciplines. One major aspect is that our own cognitive system and also higher cognitive functions of our brain are based on visual information in the form of images. When we recall memories of events in the past, we usually 'see' these events as images. Fifteen percent of the cortical region in our brain is devoted to visual processing [Hub89]. To fully understand the way in which visual data is handled, one has to understand the stages along the visual path. This is one goal of the projects in the SFB 'Neurocognition' at the CvO University of Oldenburg which mainly concentrates on the retina. Since the visual input to the retina is sampled from an array of photoreceptor cells the analogy to digitally stored images can readily be seen [Fie87, Ati92, Fie94, PB94].

From a more technical point of view, image processing plays a major role in modern telecommunication, research and entertainment. Many developments in todays medical science would be impossible without sufficient image acquisition techniques. The Hubble space telescope delivers fascinating pictures of distance stars and galaxies and makes them accessible for millions of people via the Internet. Satellite images enable us to predict the weather for the next few days, at least. Today it is almost taken for granted to import a photo into one of the common image processing programs, enhance or manipulate it in various ways and finally send a digital copy of it to a relative or friend at the other side of the world. A part of all of these applications is hardware and the associated algorithms which process image data for further usage.

Physical models of images have gained more and more importance in the last few years. Although it might not look like dealing with digital images – apart from the optics – is a very physical task, the knowledge of lattice systems and the ability to handle statistical models has enabled physicists to contribute greatly to the field of image processing. One analogy between images and classical physical models is the study of lattice systems, which is the main topic of this thesis [Bes74, GG84, PB95, Li95a].

The majority of applications can be divided into three main areas: **image compression**, **image restoration** and **feature extraction**. All of these can further be subdivided according to the methods used in the respective problem. Due to the boom in the telecommunication market, combined with the explosive growth of the Internet and the future perspectives for mobile communications and digital photography, the efficient transmission and storage of data, and in particular image data, has become one of the major challenges for scientists. To send still images and/or video *uncompressed* over the available channels or store them without any post-processing would be an enormous waste of capacities. For example a typical colour image with image dimensions of 1024x768 points would require more than 2 megabytes; however after compression with one of the standard methods this can be reduced to approximately 160 kilobytes of data. Hence a large community has devoted its research to the **compression of images**, which has resulted in several standard compression techniques, for example the well-known JPEG (Joined Photographic Experts Group [Pen90]), the proposed JPEG2000 [Say00] for still images, or the MPEG (Moving Picture Experts Group [MPG]) dealing with video data.

The existing compression methods can be divided into two groups. The first one is termed **lossy compression** and is used in most of the daily-life applications. It utilizes the fact that the human eye is insensitive to spatially higher frequencies in the image, which means that we simply don't see any rapid changes in the region of interest in the image [SJ72, RJ88]. A similar behavior can be found in our auditory system, where higher frequencies are 'masked' by lower frequencies [MN79]. Note that the decompressed image is in this case not the exact reproduction of the original data, but the difference is small enough to remain unnoticed to a human observer. A good overview of the techniques which achieve this kind of compression and encoding of the image can be found in [Say00].

The second group – **lossless compression** – is mainly used in scientific or medical applications where it is important to recover the exact information from the compressed data or for the compression of very simple images with large contiguous areas of single colours. This is especially important for medical imaging like MRI (magnetic resonance imaging) or CT (computer tomography). In fact one has to be very careful when using image compression without the proper knowledge, since recent studies have shown that lossy techniques may introduce image artefacts which can be interpreted as carcinogenic tissue [Ruj00]. Lossless compression usually achieves rates of 50 percent data reduction.

Both compression methods, lossy as well as lossless, exploit a statistical property of natural images which is called 'smoothness'. This means that the difference in intensity between adjacent pixels is not large and the neighbourhood of a point in the image can be considered a smooth surface. The basic idea of *source-coding* is to find a model for the data which describes the statistically important properties of the image, in this case the smoothness. The image can then be transformed into a representation where the differences from the model are small and the data which has to be stored is only the difference between the model image and the original image. For example, let the model be: 'All adjacent pixel have the same intensity', then we only need to store the difference from the model – which is the difference in pixel intensity. Now, since this range of actually occurring values is smaller then the intensity range itself, we need less than the number of bits per pixels. Other source coding methods are based on the property that different regions in the image are similar to other regions and can be mapped onto them via an affine transformation. This is known as fractal compression [Jac89, BH92, Fis95].

Moving away from compression we know that all transmission channels are subject to noise, which requires robust methods to remove this noise or to encode the data in a way which makes it insensitive to distortions. The source-coding idea can be used for this purpose as well. **Image restoration** aims at reducing the amount of noise in an image introduced by the transmitting channel. Again the assumption is that the image can be described by a certain model. The source image should – in the ideal case – be exactly predictable by the model. During the noisy transmission the received image is moved away from the original and hence the prediction. The restoration process will then move the image back ‘closer’ to the model and in this way reconstruct the original data. Note that in order to evaluate the restoration quality an objective error measure has to be defined. Consider again the preceeding example: if we find that a pixel after the transmission is very different from its neighbouring pixels then it is very likely that it was transmitted incorrectly. To counteract this, one could set the new (restored) value to the mean value of the surrounding values.

The art here is to find a good model which describes one given image, but is also general enough to encompass a large number of different images. This is where statistical physics comes into play, since the ‘classical’ physical model which is found in image processing techniques is derived from statistical models of particle systems. All these models belong to the class of Markov Random Fields (MRFs), which describe the interaction of neighbouring particles on a lattice by modeling the coupling between adjacent partners. One especially interesting point is that Markov Random Fields and the well-known Gibbs Random Fields are equivalent [HC71] which enables physicists to put the image processing problem into a stochastical framework which provides an in-depth knowledge of the important system properties. The pixel intensity, or any other feature of the image, can then be interpreted as the state of a particle at this point which interacts with its surrounding neighbours.

The third application is the **extraction of features** from an image. These features label certain objects of interest in a given image that need to be isolated from the remaining points. The nature of the feature that can be used depends on the available data and on the objects to be extracted. A simple example is the search for lines and edges in an image which separate different regions of intensity, which again might separate objects in the original scene. The simplest way to find those differences in intensity is by applying a gradient filter. In this way extraction and segmentation is widely used as a preprocessing step in object-recognition or movement tracking systems. It is also interesting from a biological point of view, since our own visual systems performs a similar task to distinguish objects/subjects moving around in our environment.

Despite all the advances in the modern image processing research, it is fascinating to see that the best image processing systems remains still unmatched: our own visual system.

Chapter 2

Information Theory and Learning

This chapter contains a brief introduction to information theory in general, including the definition of important quantities like entropy, information and code length and shows how model-building can be used for data compression or error correction. It is shown that learning is a consequence of providing the receiver or observer with the required statistics and the model of the data source. The introduction is completed with the motivation of how statistical physical models can contribute to this field.

2.1 Shannon Model

Although the idea of a quantitative measure of information has been around for a while, the mathematical principles which are now called information theory were established by Claude E. Shannon in 1948 [Sha48]. It is based on the simple model which is depicted in Figure 2.1 (upper part).

Before sending the data which originates from the source **S** it has to be encoded for the transmitting channel **C**. How this channel coding is performed depends on the data and on the properties of the channel, since the encoding has to be suitable for the particular transmission. The receiver **R** has to ensure that the channel encoded data is decoded correctly in order to rebuild the original data. The channel properties can be characterized by its *capacity*, which is the maximum amount of information that can be transmitted at a given time, and the *channel error*, which is the stochastic model of the qualitative and quantitative noise. With the help of this model it is possible to mathematically define the important measures *information*, *entropy*, *capacity* and *noise* within the framework of probability theory.

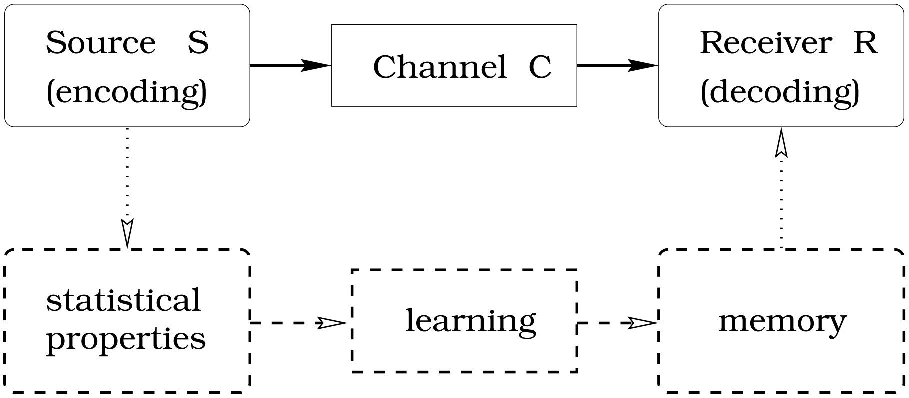


Figure 2.1: Shannon's model of an information transmitting system (upper part). The lower path is an extention to the basic model and outlines the way in which learning can provide additional information to the receiver.

2.1.1 Self-information

This measure relates the probability of an event to how much information the occurrence of this event contains. For example if the probability of an event is low, the amount of self-information associated with it is high and vice versa.

Let x be an event, from a set $\mathcal{S}$ of outcomes of some random experiment. If $P(x)$ is the probability that the event x will occur, then the self-information associated with x is given by

$$\mathcal{I}_b(x) = -\log_b P(x), \tag{2.1}$$

where b is the base of the log function. If the size of the set (number of possible events) is M and the probability of an outcome is the same for all elements, then $\mathcal{I}_b = \log_b M$ (Hartley equation). Note that the base depends on the unit of information and is not specified here. However, the unit that is used in information theory is *bits* which corresponds to the base $b = 2$. Unless otherwise noted we use this base.

2.1.2 Entropy

Entropy is the average information for a given set of events and depends on the probability distribution of these events.

Let $\mathcal{S}$ be a set of M elements and $P(x)$ the probability that the event x will occur, then the entropy or average information is:

$$H_{inf}(\mathcal{S}) = \sum_{x=1}^{M} P(x)\mathcal{I}(x) = -\sum_{x=1}^{M} P(x)\log_2 P(x). \tag{2.2}$$

The entropy has its maximum at $H_{inf} = \log_2 M$ for $P(x) = 1/M$ which means that all events have the same probability of occurrence. It is minimal if one of the events has $P(x) = 1$ and all others do not occur. Then $H_{inf} = 0$, since $P(y)\log_2 P(y) = 0$ for $P(y) = 0$.

2.1.3 Coding and source coding

As stated above, the data has to be encoded in order to be sent over the channel. By coding we mean the assignment of (binary) sequences to elements of an alphabet. The set of binary sequences is called a *code*, and the individual members of the set are called *codewords*. An *alphabet* is the collection of symbols called *letters*.

Let $n(x)$ be the number of bits in the codeword assigned to the event x, the average code length l for each code is then

$$l = \sum_{x=1}^{M} P(x)n(x). \tag{2.3}$$

Note that in the ideal case the code length is equal to the entropy. However, the code has to be *uniquely decodable* which means that the original event has to determined from a codeword. An example for a unique coding is the Huffman code [Huf52].

The **source coding** theorem relates the entropy to the average code length: Let the source contain a set of M elements x with a distribution $P(x)$, then it is possible to find a coding with an average code length l for this source with

$$H_{inf} \leq l \leq H_{inf} + 1. \tag{2.4}$$

This means that it is possible to find an encoding that can achieve a compression of the data which requires not more than one bit more than the entropy.

2.1.4 Channel coding and capacity

The capacity C is the amount of data which can be sent over the transmitting channel. It is the theoretical limit depending on the channel noise and the model for this noise. The channel coding theorem states that all transmission rates below the channel capacity are possible. This means in particular that for large blocks of data the probability of an error (in the decoded data) tends towards zero. Note that the theorem does not provide the procedure of *how* to encode the source symbols. Finding the best encoding therefore remains the challenging task.

2.2 Models provide information

In most cases it is not possible to calculate the entropy of the source exactly, since the underlying *exact* probability distribution is not known[1]. For this reason having a good mathematical model for the data can be useful for estimating the entropy of the source, which also results in a higher possible rate for compression purposes. The closer this estimation matches the true data the better it can predict the underlying process. In the ideal case where the process that creates the source data is known exactly and can be described by a model, the average information is also known exactly. In the mathematical framework of complexity theory this is known as the Kolmogorov complexity, which is the minimal algorithm (procedure) that is necessary to create the data. If this algorithm can be found or is known for the source data it is sufficient to send the algorithm itself and re-create the data at the receiver site.[2] If for example the source creates data according to a simple function $f(t), t = 1 \ldots N$ the worst case (for large N) would be to send the function value for each single t. However, if the function itself only depends on a few parameters and the class of the function is known (e.g. polynomial or trigonometric functions) the entropy of the function's code and parameters will be much smaller than the data samples.

---

[1] Estimating probabilities from high dimensional data is the topic of Chapter 7.

[2] In the light of this, the term 'source code' used in computer science can be understood as the (hopefully) minimal code which describes the final program. Unfortunately the complexity of the original code can sometimes exceed the capabilities of the application.

Where the complexity of the underlying process is too complicated for an exact model, we can obtain a model based on empirical observation of the statistics of the data.

2.2.1 Independent data

The simplest statistical model for the source is to assume that each data sample is independent of every other sample, and each one occurs with the same probability. This means that nothing about the generating source is known. However, as Equation (2.3) for the code length shows, keeping the independency assumption, but assigning the corresponding probability to each sample, results in a reduction of the transmitted data. In this case we would assign a longer code to less frequently occurring samples and the shortest code to the one with the highest probability.

2.2.2 Markov chains

The next step is to drop the independency condition and put the dependency of succeeding samples into a suitable model. A common way to do this, is by using Markov models, which describe the relation of two or more observations with conditional probabilities.

Let $x_1, x_2, \ldots x_N$ be a sequence of data samples then this sequence follows a kth-order Markov model if

$$P(x_n|x_{n-1}, x_{n-2}, \ldots, x_1) = P(x_n|x_{n-1}, x_{n-2}, \ldots, x_{n-k}). \tag{2.5}$$

This means that knowledge of the past k samples (or states) is equivalent to knowing the entire past history of the process. The very simplest model is the first order Markov chain, where $k = 1$ and the current state only depends on the previous one. Using Markov models in this simple way however, would require the sampling or storage of all possible conditional probabilities. For images that have an intensity resolution of $x = 0 \ldots 255$ levels (grey levels or states per pixel) the simplest model with $k = 1$ already needs a table of 65536 entries for each pair $P(x_n|x_{n-1})$. Hence, it is necessary to find a model which describes these probabilities with a function that depends on only a few parameters. There are known models from statistical physics that exactly meet this requirement.

2.2.3 Statistical physics

The interesting point for physicists is that the Markov models which are used in image processing are related to some statistical, physical systems in a special way. Namely, if the neighbouring points on a lattice are connected by conditional probabilities of a Markov field then these probabilities can be described by a Gibbs distribution. This means that the conditional probabilities of two (or more) related samples can be written down by using a local energy and potential function which couples these two samples. A Gibbs distribution takes the following form

$$P(s) = \frac{1}{Z} e^{-E(s)/T}, \tag{2.6}$$

where

$$Z = \sum_s e^{-E(s)/T} \tag{2.7}$$

is the partition function, T is the temperature and $E(s)$ is the energy function

$$E(s) = \sum_i V_i(s) \tag{2.8}$$

with the sum over all local potentials. The form of the energy and the related properties will be explained in more detail in Chapter 4. These energy functions usually contain only a few parameters, yet provide a powerful method for modeling the complexity of natural images.

2.3 Learning and memory

The knowledge of the underlying model which generates the data at the source can be used to either reduce the amount of transmitted data or to correct erroneous transmissions. In a biological system, learning can take place during the evolution of the whole species, in which case the (visual) system develops the necessary 'hardware' to process the data. It can also take place in the earlier stages of life with the growth of neural tissue [Hub89]. Since the learning is inherently unsupervised, except for the fact that the individual has to detect a potential predator or find some food in order to survive, the model has to be learned from the statistics of the data alone. This process is displayed in Figure 2.1 (lower path) as an extension to the simple information transmition system. The statistical properties of the source data are learned, stored in memory and can be recalled at the receiver site. In the case of the visual system, the source is the visual input, learning is the process of evolution (or adaptation in childhood) and memory is the knowledge of the statistics of the model (the prior) [Bar89]. If this assumption is correct then the visual path up to the visual cortex should have an architecture that provides exactly this kind of processing. It is known that at least in the retina the layers of neurons work as spatial filters to the visual input, which are sensitive to local contrast and can adapt to the overall intensity. One of the experiments in Chapter 6 uses a simple model to demonstrate the importance of local correlations between points in the image by breaking up these correlations. This results in a perceivable degradation of the image.

2.4 Bayesian methods

These methods are part of the field known as *Bayesian probability theory*, which is named after T. Bayes, an 18^{th} century mathematician. Its main result is the Bayes theorem which will we used in all the following chapters. The main concept in Bayesian theory is that all the probabilities involved are *conditional*. This means that the probability depends on the evidence of the event or in a more formal definition: given an event A with nonzero probability $P(A)$, the conditional probability of B given A is

$$P(B|A) = \frac{P(B \cap A)}{P(A)}, \tag{2.9}$$

where $P(B \cap A)$ denotes the event where both A and B occur. If they are both mutually exclusive, then $P(B|A) = 0$. The interesting question is now, how to invert this conditional probability, to find $P(A|B)$. This is stated in the Bayes theorem (also known as Bayes inversion)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{2.10}$$

The probability $P(A|B)$ is called *a posteriori probability* and $P(A)$ is the *a priori probability* of A (prior). The denominator is the sum over all events $P(B) = \sum_{A_i} P(B|A_i)P(A_i)$. In the physical formalism used in this work, the *a priori* probability is the Gibbs distribution, which describes the properties of the source image and the conditional probability $P(B|A)$ contains the stochastic model of the noise. The Bayes formalism hence requires a statistical model for source and the transmitting channel.

A common way to find the best restoration for the data is to demand that the *a posteriori probability* is maximized (maximum a posteriori, MAP estimate), which gives the most probable event A (image) given the data B (received data). In a more general approach the optimal estimate for a restoration can be found by minimizing the Bayes risk which is defined as

$$R(\hat{s}) = \int_s C(\hat{s}, s) P(s|r) ds, \tag{2.11}$$

where r is the (received) data, $C(\hat{s}, s)$ is a cost function and $P(s|r)$ the posterior distribution. The cost function determines the cost of the estimate $\hat{s}$ when the truth is s and can be chosen according to the problem and/or requirements for the error. Since this is the restoration error of the image it should take into account a model of the subjective error of an observer. Minimizing the risk will then lead to the optimal image. Note that finding a suitable optimization procedure is in itself a difficult task.

Chapter 3

Image and error statistics

Before starting to think about a model it is necessary to understand the properties of the data to be modeled, especially the statistics. In the case of image processing this means finding a way to describe the class of 'natural' images, but also the statistics of the noisy transmission channel. This chapter presents an analysis of a set of natural images, extracts the important statistical attributes and describes the types of noise which can be found in common problems.

3.1 Natural images

Images originating from a natural source have characteristic properties which distinguishes them from 'random' images. The grey level images that are used in the following models are digitized and stored in a computer as two-dimensional arrays where each field of this array has an integer value in the range of 0 to 255. This value is proportional to the brightness of the small area in the original image at this point. Note that brightness is not the same as intensity. Intensity is proportional to the area and the incident power within the range of the electromagnetic spectrum. There are two reasons why brightness is stored instead of intensity: first, the intensity I_{screen} displayed by the cathode-ray tube of the computer screen relates to the computer's voltage-signal U as

$$I_{screen} = const. \times U^{\gamma}, \tag{3.1}$$

where a typical value for γ is around 2.2. Second, brightness is a subjective measure and has experimentally been found to relate to intensity as :

$$brightness = const. \times intensity^{0.3}. \tag{3.2}$$

Since the eye perceives a logarithmic scale, it is not efficient to sample the intensity linearly down to 256 values. For this reason a so-called *gamma correction* is used, which means that the intensity is transformed by the inverse function of the computer screen and this value is stored. The eye's brightness law Equation (3.2) is approximately the inverse of the screen Equation (3.1) and the result is an optimal sampling of the original brightness.

A colour image often consists of three colour planes for the red, green and blue component respectively and each of these planes can be treated as a monochromatic image with 256 levels of brightness for each pixel, adding up to 24 bit per pixel for the full colour version. This representation is similar to that found in the human retina which consists of four different types of receptor cells, for different ranges of wavelengths, placed on a non-regular grid. The *rods* are responsible for non-colour vision, whereas the three types of *cones* are sensitive to the colours red, green and blue.

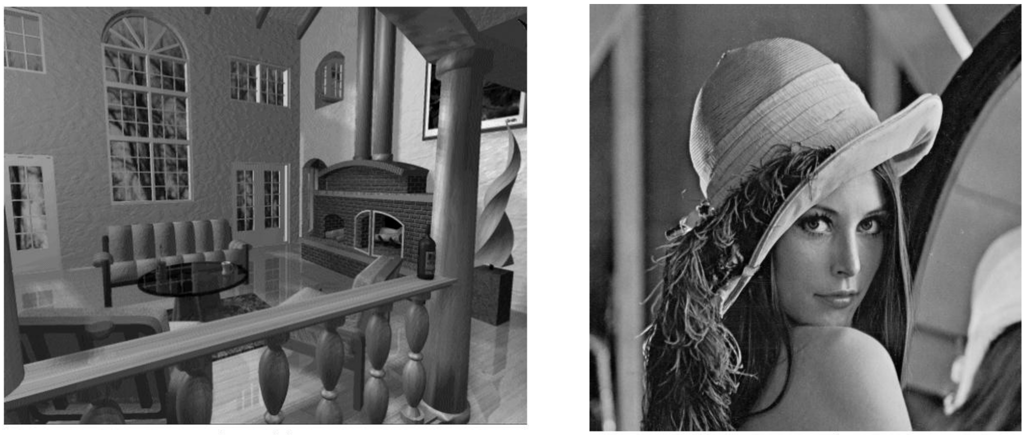
The difference between a random image in which each pixel is assigned a random value drawn from a uniform distribution and a 'natural' image sampled from a real scene is the strong correlation between neighbouring points. The brightness difference between two adjacent pixels in a natural image is on average one order of magnitude lower than the maximum brightness. This restricts the number of accessible points in the state space of all possible configurations to a small region.

3.2 Image statistics

As stated above, natural images are scenes from the 'real world' captured by an image acquisition system like a camera and a scanner or a digital camera or other more specialized devices. The test set of the three images displayed in Figure 3.1 is used to investigate the properties of natural scenes and compare them to the random image given as the fourth candidate. The sources of the images are displayed in Table 3.1.

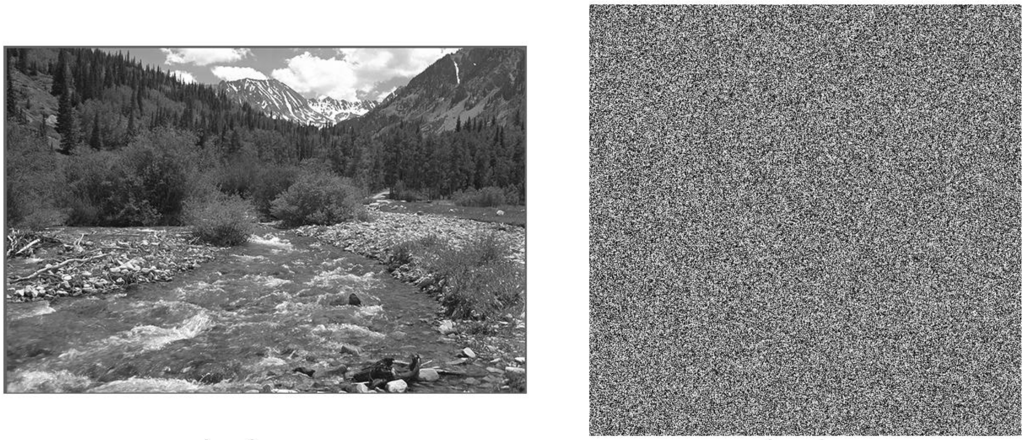
3.2.1 Zero-order statistics

As stated in Section 2.2.1, the first and simplest model for a data source which has a non-uniform distribution of sample values is to estimate the probability for each sample and assign a longer code to less probable values. The histograms in Figure 3.2 contain the frequency as an approximation to the probability that a given point takes this brightness, or in short:the



rendered image

portrait



landscape

random image

Figure 3.1: The four images used for illustrating some of the statistical properties of image data. Upper left: computer graphic created with ray-tracing and rendering. Upper right: portrait. Lower left: landscape (both scanned from photograph) Lower right: random image with each pixel set to a random value.

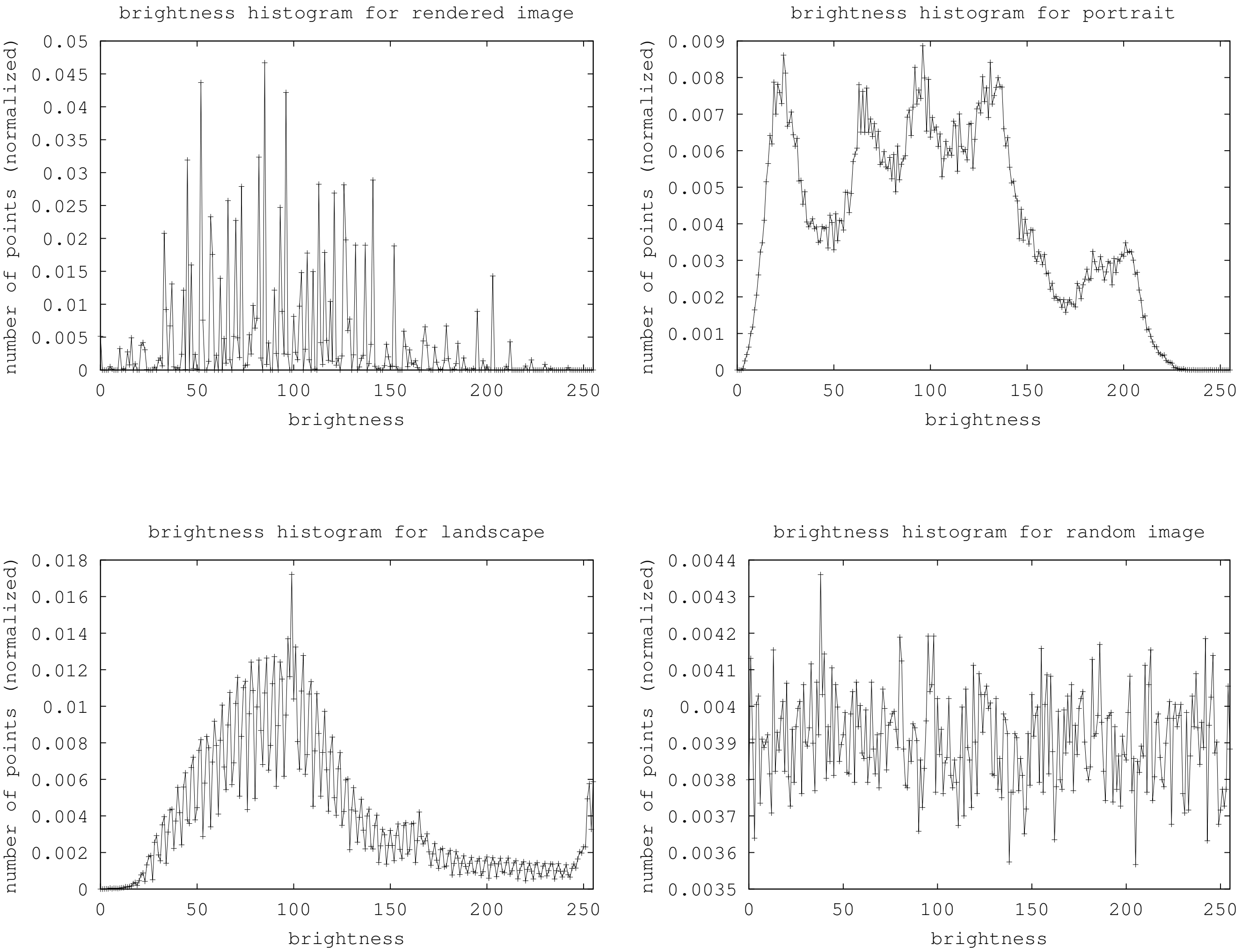


Figure 3.2: Distribution of brightness levels in the images (from Figure 3.1). Due to the varying number of pixels between the images, the frequency is normalized.

image	dimensions	number of brightness levels	
1 (ul)	600×480	151	rendered computer graphic
2 (ur)	512×512	236	scanned photo (portrait)
3 (ll)	768×512	254	scanned photo (landscape)
4 (lr)	512×512	256	random image (uniform distribution)

Table 3.1: Characteristics of the set of test images in Figure 3.1

brightness distribution of the image. The computer-generated image has an irregular distribution, since it contains only 151 of the 256 possible states. Both the portrait and the landscape show one or several distinct peaks in the histogram in contrast to the random image, which is uniformly distributed.

3.2.2 First order statistics – Markov model

The next step towards a more complex model is the use of conditional probabilities in a first order Markov model. Here the value of the next sample is predicted by the current sample. In this case the model is simply: the next value is equal to the current one and only the difference from the prediction has to be encoded. The interesting quantity to look at is the distribution of differences in brightness of two neighbouring pixels. For each point in the image the brightness of this pixel is subtracted from the horizontal neighbour (in the x-direction) and put into the histograms displayed in Figure 3.3. In these histograms the main property of natural images can already be seen: all distributions have a sharp peak around zero (no difference) and large difference have a very low probability – the images are smooth. The shape of the distribution can be approximated by a function with only a few free parameters which already captures the whole complexity. These functions usually have a Gaussian shape [Li95a, Say00], but it was found that a lorentz function gives a better fit to the data:

$$f(x) = \frac{a}{b^2 + (x-c)^2}, \tag{3.3}$$

where $c \approx 0$. Again the artificial image has a slightly different behavior in that a fraction of 0.3 of all neighbouring pixels are identical (constant surface) and some of the larger differences occur due to the irregularity in the brightness distribution. It should be noted here that the retina works only on differences in brightness and adapts to the global brightness level.

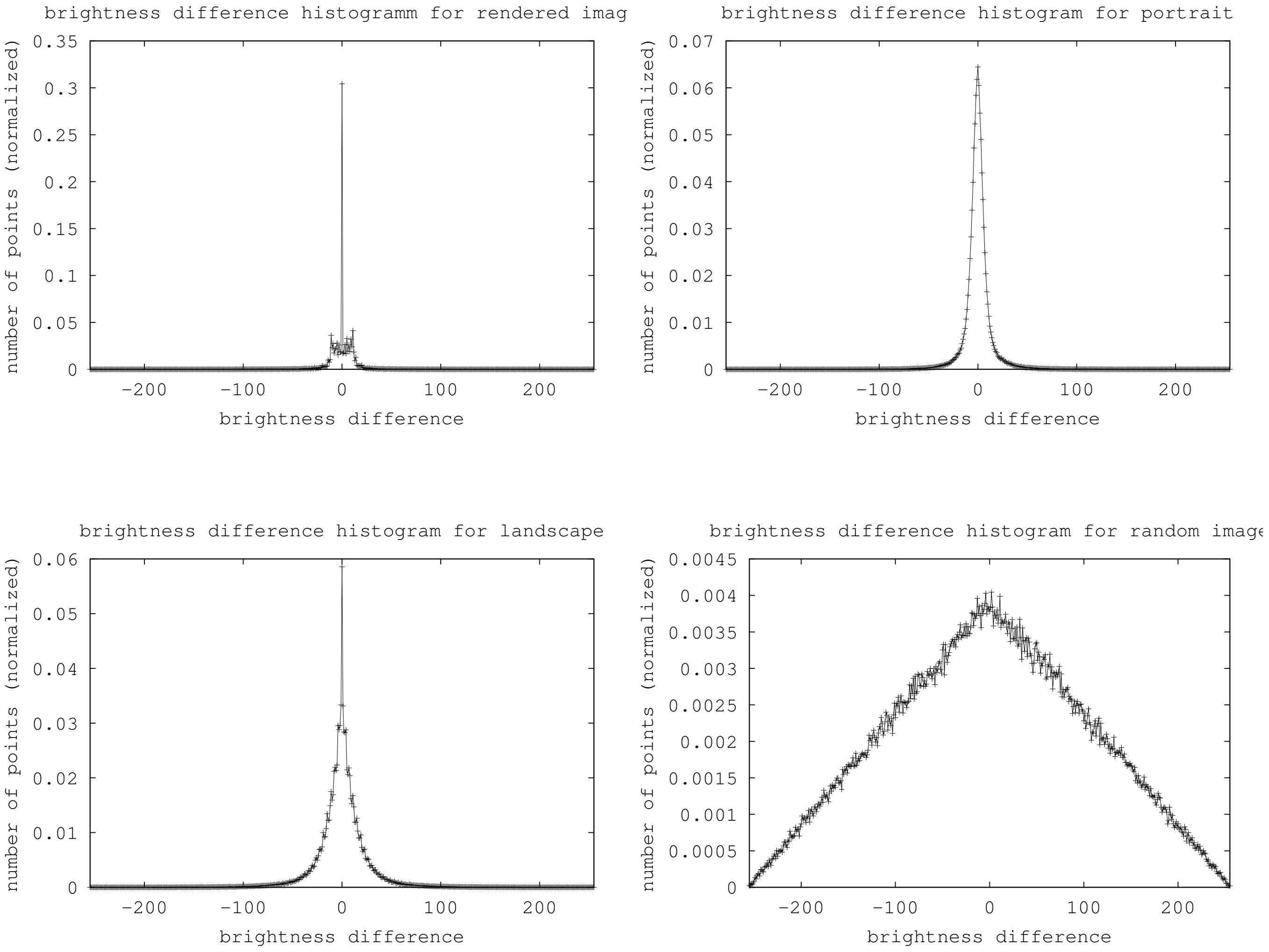


Figure 3.3: Distribution of brightness level difference between two neighbouring horizontal points in the images (from Figure 3.1). The frequency is normalized.

3.2.3 A quick glance at the Zq-model

The model that will be examined later in Chapter 5 is based on the absolute difference of two neighbouring pixels and can be termed a contrast model. It captures the full range of correlations between these two points. This approach is illustrated in Figure 3.4, where the correlation strength (amplitude) is plotted depending on the difference between pixels (wavenumber). Mathematically, the plots are the discrete cosine transformations of the distributions in Figure 3.3. The curves belonging to the portrait and landscape image show a similar behavior, however the later has a local peak around the wavenumber 140 which is a result of the local fluctuations in brightness in the scene (the small stones on the shore, waves on the water and leaves). Again the rendered image reveals its artificial origin in the strong oscillations in the spectrum.

3.2.4 Entropy

In order to give an estimation of how much information is contained in the model compared to the original non-encoded source data, the entropy for the different models is calculated using Equation (2.2). The results are listed in Table 3.2.

image	zero-order in bit/pixel	1-order in bit/pixel	Zq-model in bit/pixel	CALIC model in bit/pixel
rendered (ul)	6.124	4.767	4.133	4.264
portrait (ur)	7.594	5.328	4.401	4.552
landscape (ll)	7.431	6.339	5.834	5.920
random (lr)	8.000	8.720	8.277	10.022

Table 3.2: Entropies of the set of test images in Figure 3.1 for different models.

The uncompressed code length for one pixel in the images requires 8 bit/pixel. Compared to this the zero-order model does not contain much information. The 1-order and the Zq-model perform significantly better, which shows that much of the image statistics is already contained in the short range correlation between two points. This does not work for the random image, since in this case the dynamic range per pixel requires the values from -255 to 255 (difference between the pixels). As an estimate for the entropy of the image – although this might be far away from the true value – the results from the currently best lossless compression method CALIC (context-based,

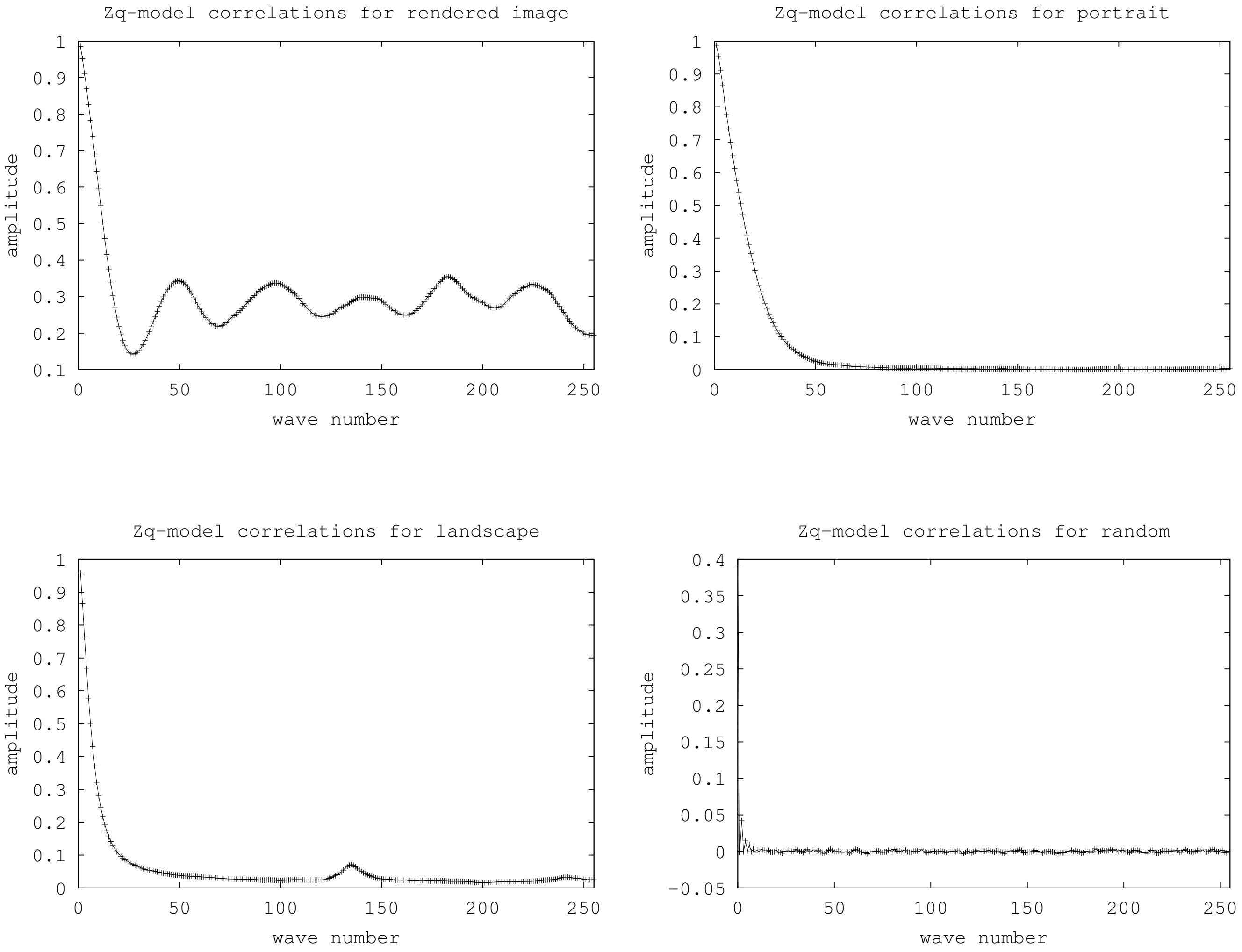


Figure 3.4: Correlations obtained from the Zq-model, again measured in the set of test images (Figure 3.1). The correlations are taken between two horizontally neighbouring points.

adaptive, lossless image coder) [WM96] are listed for comparison. This is based on a larger neighbourhood of pixels and adapts the parameters of the model to the current image region. However, the performance is very poor with the random image, because it violates the model incorporated in the context-encoder.

3.3 Common image models

To conclude this section about image statistics, a short overview of common image compression algorithms is presented. This is to illustrate what properties of natural images can be used for creating a model.

Predictive coding has been used in the previous sections; it is based on the idea of estimating a pixel value from the previous or surrounding samples. This works well for many images since neighbouring pixels often have strong correlations [WM96, Rus99, Say00]. Other compression schemes do not aim at reconstructing the original image exactly, but the decoded image should be as close as possible (given an error measure) to the original. One of these methods is **transform coding**, which decomposes the image (or blocks of the image) into components which can be further encoded according to their characteristics. Several methods use a transformation that de-correlates the components and then encodes the components of the new representation separately. Members of this class are discrete sine- and cosine-transforms (JPEG) and principal component analysis [Jai81, CMZ89, Say00]. An application for feature extraction using this methods is presented in Chapter 7. Another method is based on **wavelet transformations** which decompose the data on multiple spatial scales. Here a single function forms the base for the other components which are scaled and translated versions of the 'mother(wavelet) function'. Although related to sine and cosine transform this has the advantage of representing the image on different scales [Say00]. **Vector quantization** encodes several samples (in an image block) at the same time by assigning a code-vector as a representative to the block which has the smallest (euclidian) distance. The code-vector is kept in a code-book and addressed by an integer index. Instead of sending the whole vector, only the index of a block and the code book has to be sent once for a whole image [Jai81, NK88, KK96, Abu90]. The novel approach of **fractal compression** can be seen as the search for a fixed-point transformation that has the final image as the fixed-point [BH92]. Instead of encoding and sending the image, the function generating the data is sent. A solution to this inverse problem was suggested in [Jac89] where the image is partitioned into smaller blocks and a geometric transformation maps these blocks to another smaller image region.

3.4 Noise

So far the investigated properties were that of the *prior* source data. However, for the transmission model it is also necessary to understand the nature of the channel noise. This requires suitable models for the noisy channel. This noise can originate from several sources: electronic noise during a transmission, atmospheric disturbance in long range transmissions, interference between different parts in circuits or biological noise due to fluctuations in action potentials and thermal movements, but also optical blurring or out-of-focus imaging. All of these can be subdivided in two classes: stochastic uncorrelated noise, and deterministic filters. Although one might expect that a deterministic mapping (like a linear filter) is mathematically exactly invertible and hence does not contribute to information loss, this is in fact not the case. Additionally, all source models rely on an accurate estimation of the underlying probabilities. However the number of data samples is limited to the size of the image and the sampled frequencies only approximate the exact probabilities. This results in finite-size effects, which can in principle be estimated from the data [Gra88].

The general model for a noisy transmission is expressed in the equation

$$r_i = D(\vec{s}) + \eta_i, \tag{3.4}$$

where $\vec{s}$ is the whole set of samples, $D(\vec{s})$ is an arbitrary function of this set and η_i an additive, stochastical noise. If we restrict this model to linear, finite filters this simplifies to

$$r_i = \sum_{j=-n}^{n} d_j s_{i+j} + \eta_i, \tag{3.5}$$

with d_j as the linear filter coefficients and n the size of the filter window. The following sections provide a description of the types of error examined in the experiments.

3.4.1 Stochastic noise

Stochastic noise is uncorrelated and independent of the data and sample (image) position. Usually it adds a random value drawn from a distribution to the sample value or it changes the value into a new value with a certain probability. This last type is called **random noise**, it can introduce pixels of high brightness in darker regions and dark pixels in bright areas.

$$r = \left\{ \begin{array}{ll} s & \text{with probability } (1-p) \\ z & \text{(where } z \neq s \text{) with } p/(q-1) \end{array} \right. , \tag{3.6}$$

where p is the probability of changing the original pixel s during transmission into any other possible value z not equal to s. Here q is the number of all possible values (in an image this would typically be 256). The conditional probability of receiving r given the original s is

$$P(r|s) = (1-p)\delta_{r,s} + \frac{p}{q-1} \sum_{z \neq s} \delta_{r,z}, \tag{3.7}$$

where $\delta_{r,z}$ is the Kronecker delta. This can be written as an exponential function

$$P(r|s) = e^{a + b\delta_{r,s}}, \tag{3.8}$$

with

$$a = \ln\left(\frac{p}{q-1}\right) \tag{3.9}$$

and

$$b = \ln(1-p) - \ln\left(\frac{p}{q-1}\right), \tag{3.10}$$

which can be interpreted as an external field of strength b that couples the original and received image. The stronger the coupling, the smaller the noise. The information loss for this type of noise can be calculated using the entropy

$$\begin{aligned} H_{random}(p) &= -(1-p)\log_2(1-p) - \sum \frac{p}{q-1} \log_2\left(\frac{p}{q-1}\right) && (3.11) \\ &= -(1-p)\log_2(1-p) - p\log_2\left(\frac{p}{q-1}\right), && (3.12) \end{aligned}$$

where the sum runs over $q-1$ states. In the case of $p = 1$ and $q = 256$ the information loss is $H_{random}(1) = \log_2(255)$, which is not exactly 8 bit, since the receiver at least knows that the received value is *not* the source value.

A related type of noise is **salt-and-pepper** or **impulse noise** which adds or subtracts a fixed value z to or from the original sample.

$$r = \begin{cases} s & \text{with probability } (1-p) \\ s+z & \text{with probability } p/2 \\ s-z & \text{with probability } p/2 \end{cases} \quad . \tag{3.13}$$

The conditional probability is

$$P(r|s) = (1-p)\delta_{r,s} + \frac{p}{2}\delta_{|r-s|,z} \tag{3.14}$$

and in exponential form

$$P(r|s) = e^{a\delta_{r,s} + b\delta_{|r-s|,z}}, \tag{3.15}$$

with

$$a \quad = \quad \ln(1-p) \tag{3.16}$$

and

$$b \quad = \quad \ln\left(\frac{p}{2}\right). \tag{3.17}$$

The information loss gives

$$H_{imp}(p) = -(1-p)\log_2(1-p) - p\log_2\left(\frac{p}{2}\right). \tag{3.18}$$

Another type of noise is the additive **Gaussian** or **white noise**. In this case a random value drawn from a normal distribution $N(\mu, \sigma)$, where μ is the mean value of the distribution (which is almost always taken to be zero) and σ the variance, is added to each sample value. The conditional probability for $\mu = 0$ is

$$P(r|s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r-s)^2}{2\sigma^2}\right), \tag{3.19}$$

and the information loss (from [CT91]) is

$$H_{white}(\sigma) = \log_2 q - \frac{1}{2}\log_2\left(1 + \frac{q^2}{\sigma^2}\right). \tag{3.20}$$

3.4.2 Correlated noise – filters and blurring

Correlated noise described by the linear filter Equation (3.5) does not introduce any stochastical distortions (as long as $\eta_i = 0$). However it correlates neighbouring pixels by assigning each one a linear combination consisting of the adjacent values multiplied by the filter coefficient. This can be written down in a vectorized form for the whole image as

$$\vec{r} = \mathbf{D}\vec{s}. \tag{3.21}$$

This simple model is sufficient for most problems encountered in image processing. In most cases this type of degradation results from imperfect optics, a non-focused system or motion blur. The filter coefficients and the shape of the filter have to be determined for these cases. Note that another way of representing a linear filter is by a convolution of the original data with a corresponding convolution function. Deconvolution is typically achieved by using the Fourier transform of these two and dividing this transform of the data by the the transform of the filter kernel, which is equivalent to determining the inverse of the filter matrix D. Now if the transformation D is invertible, then it is possible to recover the original s exactly. But this is true only in mathematical terms! Namely, the received values must have an arbitrary precission or at least a sufficiently high precession for the transformation. However, since the original image has values in the integer range of 0 to 255 the received image will have the same range and this can only be achieved by quantization. This quantization process constitutes the true information loss. This is illustrated in the following example.

Assume the linear filter of size $2n+1$ has the normalized filter coefficients $d_i \in \mathbb{Q}$ (rational numbers), where

$$\begin{aligned} d_i &= \frac{D_i}{C} && (3.22) \\ C &= \sum_{j=1}^{2n+1} D_j, && (3.23) \end{aligned}$$

with $D_i \in \mathbb{N}$ and $C \in \mathbb{N}$. Given the data $s_i \in \{0, 1, \ldots, 255\}$ the received values are

$$\begin{aligned} r_i &= \sum_{j=-n}^{n} d_j s_{i+j} && (3.24)\\ &= \sum_{j=-n}^{n} \frac{D_j s_{i+j}}{C} && (3.25)\\ &= \frac{1}{C}\sum_{j=-n}^{n} D_j s_{i+j} && (3.26)\\ &= \frac{R_i}{C}. && (3.27)\end{aligned}$$

Now in order to keep the range of r_i in the domain of the 8 bit image values, each r_i has to be an integer and remain within the set of allowed values $\{0, 1, \ldots, 255\}$. Since the configuration of the s_i is completely arbitrary, this is not possible in general. In this case it is necessary to quantize the data back to the original range. This means the value $R_i \in \mathbb{N}$ is divided by C in an integer division where the remainder is simply discarded. Now if R_i is a multiple of C no information is lost. But in the worst case the remainder is $C - 1$. To ensure that no information is lost one would have to send this remainder with $\log_2(R_i \mod C)$ bits in addition to the value of r_i. This can be interpreted as sending an additional 'modulo image' which encodes the remainders. The entropy of this image represents the loss of information, or more precisely the information which has to be sent in addition to the data in order to reconstruct the true unfiltered value. Note that the two entropies are not strictly additive, for large C the modulo image can require more bits per pixel than the data image.

A second problem can be seen from the deconvolution. In the mathematical (simplified) notation the filtering can be written as

$$R(x) = \int S(y)D(y-x)dy, \tag{3.28}$$

which in the Fourier space is simply the product

$$\mathcal{R}(k) = \mathcal{S}(k)\mathcal{D}(k). \tag{3.29}$$

Now deconvolution is achieved by dividing with the filter

$$\mathcal{S}(k) = \mathcal{R}(k)/\mathcal{D}(k). \tag{3.30}$$

Obviously, the Fourier transformation of the filter may not be zero in the frequency domain, otherwise the ratio is undefined. This problem can be circumvented by setting these points to small values [PTVF92].

For the Bayesian restoration the remaining problem is to put the linear filter into a statistical model. Now since the filter is deterministic the conditional probability is

$$P(r_i|s_i) = \begin{cases} 1 & \text{if } \left(Cr_i = \sum_{j=-n}^{n} D_j s_{i+j}\right) = 0 \\ 0 & \text{else} \end{cases} . \tag{3.31}$$

In order to use this in the restoration model the conditional probability has to be inverted according to the Bayes Equation (2.10) to obtain the dependence of $P(s_i|r_i)$. Since there are several configurations of neighbouring values which can result in the same r_i when summed, the probability is not only simply equal to 1 if r_i is the filtered sum of the s_{i+j}, but in general

$$P(s_i|r_i) = \frac{1}{\sqrt{2\pi\sigma_{mc}^2}} \exp\left(-\frac{1}{2\sigma_{mc}^2}\left(\sum_{j=-n}^{n}\left[r_{i+j} - \sum_{k=-n}^{n} D_k s_{i+j+k}\right]\right)^2\right). \tag{3.32}$$

Note that s_i is actually the current value during restoration. The exponential Gaussian is used here in the restoration procedure for technical reasons. In this form it is easier for the Monte Carlo algorithm (see Section 4.4.4) to converge to the best value slowly. σ_{mc} is adapted accordingly.

3.4.3 Estimating noise from the data

An interesting point for a real-life transmission is the estimation of the noise if this is not known or cannot be measured. If there exists a model for the source data and it is possible to estimate all parameters entering this model, then by sending these parameters (however undistorted) the noise can be measured. To do this, the received data is checked against the source model and the difference can be used to find a model for the noise and in the best case to get an approximation for the noise strength. In the simplest case one compares the difference between the data sample and the prediction by the model. Any noise which violates this model can then be found from the measured distribution. This procedure is demonstrated in the experiments in Section 5.7.

3.5 Image quality

When talking about the quality of an image or about the performance of a model then it is necessary to have a quantitative and objective measure of the error between the true image and the restored or compressed one.

The problem is that image quality is a subjective measure and not easy to describe with a simple function. Although there exists some approaches to describe the human visual system in this respect, a good function remains to be found. Some of the suggestions have found a way into the specification for video processing [ANS96]. Still, since it is inherently dependent on the person performing the comparison, it is not clear if a general form exists. Experiments have shown that the simple measures given below correlate well with subjective perception [BP98].

The two currently accepted measures are based on the difference distortion. These are the *mean squared error* per pixel

$$d(\vec{s},\vec{r}) = \frac{1}{N}\sum_{i=1}^{N}(s_i - r_i)^2 \tag{3.33}$$

and the *average absolute error* per pixel

$$d(\vec{s},\vec{r}) = \frac{1}{N}\sum_{i=1}^{N}|s_i - r_i|, \tag{3.34}$$

where the former is used in the experiments.

Chapter 4

Markov Random Fields

This chapters contains the fundamental definitions and properties of Markov Random Fields and links them to the formalism of Gibbs Random Fields. This is then extended to image processing and restoration methods.

4.1 Introduction

The theory of Markov Random Fields (MRFs) is part of the larger field of probability theory and describes the statistical properties of (physical) systems by modeling the local interactions of system variables [GG84, Li95a, PB94, PB95]. Local in this case means that the observables are connected to each other in a neighbourhood system defined for the topology of the whole system. The aim is to simplify the system's form and complex behavior by using strictly local, simple models which require fewer parameters to deal with. From the point of probability theory, the 'only' requirement is to find the *a priori* probabilities, plus parameters which assign higher probabilities to the more probable states. Unfortunately, finding these probabilities is not an easy task. However, due to an equivalence between Markov Random Fields and the Gibbs Random Fields (GRFs) it is possible to define these probabilities through clique-potentials (local couplings) containing physical quantities like energy, coupling constants and temperature. This allows us to exploit the well studied systems found in statistical physics for image processing purposes [Bes74, Sav80].

Markov Random Fields have found a wide range of applications from image restoration [GG84] to motion estimation [KD92]. The following overview introduces the definitions and formalism used in the context of MRFs.

4.2 Markov Random Fields and Gibbs Random Fields

Digital images are in most cases stored as two-dimensional arrays, with each field representing a pixel which is the brightness at the spatial position. Hence, the system is described as a regular, discrete lattice, where each lattice point can take on one of several states. The number of states is typically finite, but in other applications a continuous state variable might be allowed. This leads to the definition of the topology of the lattice. The lattice (or grid) $\mathcal{G}$ is used to index a finite set of N elements (the pixels). It has a two-dimensional topology, which means the set is 'ordered' and follows the condition:

$$\mathcal{G} = \{(x, y) | 1 \leq x \leq n_x, 1 \leq y \leq n_y\}, \tag{4.1}$$

where $n_x, n_y \in \mathbb{N}$ are the dimensions in x and y direction respectively and $N = n_x \times n_y$ is the image size. The state of a lattice point is described by a label which in general is a member of the set of all possible labels $\mathcal{L}$ (for example $\mathbb{R}$ or $\mathbb{R}^m$). For the purpose of image processing

$$\mathcal{L} = \{l_1, l_2, ...l_q\}, \tag{4.2}$$

where l_i is the brightness level and q the number of levels (in the case of 8-bit images, $q = 256$). An important property of $\mathcal{L}$ is that the states are ordered and a relation like $l_1 < l_2 < ... < l_q$ exists.

To combine the lattice definition and the labeling, each point on the lattice $\mathcal{G}$ is assigned one state in $\mathcal{L}$ to give the system state which finally represents the image $\mathbf{s}$. In the following we use

$$\mathbf{S} = \{S_1, ..S_N\} \tag{4.3}$$

to denote *all* possible states of the image and

$$\mathbf{s} = \{s_1, ..s_N\} \tag{4.4}$$

to define *one* state (realization) of the image, that is one point in the space state of all lattice labelings. $\mathbf{S}$ can also be seen as a mapping from $\mathcal{G}$ to $\mathcal{L}$.

As has been noted above, Markov Fields are described by a system of neighbourhoods, namely

$$\mathcal{N} = \{\mathcal{N}_i | \forall i \in \mathcal{G}\}, \tag{4.5}$$

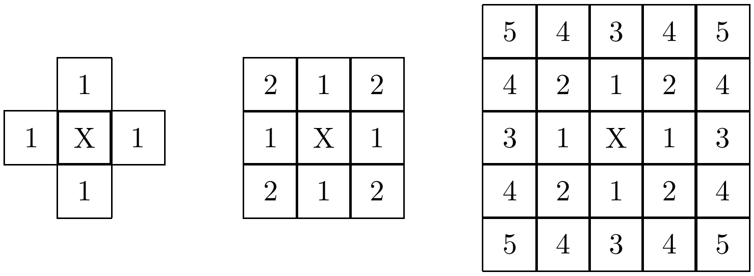
where $\mathcal{N}_i$ is the set of all neighbouring points of i and the neighbourhood relation is

1. $i \notin \mathcal{N}_i$ that is, a point is not a neighbour of itself and
2. $i \in \mathcal{N}_i' \Leftrightarrow i' \in \mathcal{N}_i$, i.e. if a point i is neighbour of i', then the point i' is a neighbour of i.

For a two-dimensional grid the neighbourhood is defined by the environment of the point

$$\mathcal{N}_i^{2D} = \{i' \in \mathcal{G} | \, \|(x,y)_i - (x,y)_{i'}\|_2 < r, i \neq i'\}, \tag{4.6}$$

which is called an n-order neighbourhood in the case of a rectangular grid. Some of the often used neighbourhood systems are displayed in Figure 4.1.



(a) (b) (c)

Figure 4.1: Neighbours on a rectangular grid. (a) first order neighbourhood, (b) second and (c) fifth order, X is the center point, the numbers are the neighbours.

4.2.1 Definition of a Markov Random Field

Let $\mathbf{S} = \{S_1, ..., S_N\}$ be a set of random variables, defined on the lattice $\mathcal{G}$, with each variable S_i taking a value s_i out of the set $\mathcal{L}$; $\mathbf{S}$ is then called a random field. For the discrete set $\mathcal{L}$, the quantity $P(S_i = s_i)$ (in short $P(s_i)$) denotes the probability that S_i takes the value of s_i. The total probability of the whole system is $P(\mathbf{S} = \mathbf{s}) = P(S_1 = s_1, ...S_N = s_N)$. $\mathbf{S}$ is called a Markov Random Field on $\mathcal{G}$ with a neighbourhood system $\mathcal{N}$ if

$$P(\mathbf{s}) > 0, \, \forall \, \mathbf{s} \in \mathbf{S} \qquad \text{(positivity)} \tag{4.7}$$

$$P(s_i | s_{\mathcal{G}\backslash\{i\}}) = P(s_i | s_{\mathcal{N}_i}) \quad \text{(Markov condition)}, \tag{4.8}$$

where $\mathcal{G}\backslash\{i\}$ is the set of all points without i, $s_{\mathcal{G}\backslash\{i\}}$ are the values or labels of the points and $s_{\mathcal{N}_i}$ are the values of the neighbours. $P(a|b)$ is the conditional probability of a given b. The reason that P is greater than 0 can be seen in the definition of the GRFs. The Markov condition means that the global system state $P(\mathbf{S})$ is described by local, conditional probabilities of the points. The state of a single lattice point depends on its neighbouring points only.

4.2.2 Definition of the Gibbs Random Field

A random field $\mathbf{S}$ is called a Gibbs Random Field (GRF) on the lattice $\mathcal{G}$, with the neighbourhood system $\mathcal{N}$ if the system configuration of this lattice follows a Gibbs distribution, of the form

$$P(\mathbf{s}) = Z^{-1} \times \exp\left(-\frac{1}{T} E_{global}(\mathbf{s})\right), \tag{4.9}$$

where

$$Z = \sum_{\mathbf{s} \in \mathbf{S}} \exp\left(-\frac{1}{T} E_{global}(\mathbf{s})\right) \tag{4.10}$$

is the partition function and T the temperature. The energy

$$E_{global}(\mathbf{s}) = \sum_i E(s_i, s_{\mathcal{N}_i}) \tag{4.11}$$

is the sum of the local energies E (or potentials) over all points and neighbourhoods. The value of E depends on the local configuration of the point and its neighbouring points. T is a control parameter influencing the ‘sharpness’ of the distribution. For high T all configurations have equal probabilities. Note that in the model introduced in the next chapter, T is not given explicitly, but is contained in the coupling constants. To calculated the partition function Z exactly, it is in general necessary to sum over all possible realizations of the system. Since this is computationally intractable, various methods can be used to approximate this quantity (see for example [Li95a]). As can be seen from Equation (4.9) the probability of a configuration which is denoted by $P(\mathbf{s})$ depends on the energy E. The lower the energy for a state the higher the probability of finding the system in this state. Hence, the model and its parameters have to be chosen to yield low energy values for more probable configurations.

4.2.3 Equivalence of Markov and Gibbs Random Fields

From a mathematical point of view MRFs are characterized by the local Markov condition Equation (4.8) and GRFs by the global property of the

Gibbs distribution Equation (4.9). However, the equivalence of these two properties is established in the famous Hammersley-Clifford theorem: $\mathbf{S}$ is an MRF on $\mathcal{G}$ with a neighbourhood system $\mathcal{N}$ if and only if $\mathbf{S}$ is a GRF on $\mathcal{G}$ with a nearest neighbourhood Gibbs potential [HC71, Bes74, Li95a].

The main benefit of this equivalence is that it provides a simple way to specify MRFs by specifying potentials instead of local characteristics, which is usually very difficult. These potentials contain the interactions or coupling of neighbouring points. However, nothing is said about the shape of these potentials, let alone the selection of the parameters involved.

4.3 Markov Random Fields and image models

MRF models in computer vision have become popular since the famous paper of S.Geman and D.Geman on image restoration [GG84]. Based on the properties described in the last sections, this makes it possible to use MRFs and GRFs in a comparatively simple way, yet providing the mathematical and physical foundation. The field has grown rapidly in recent years and has addressed a variety of image processing tasks. As well as object recognition, applications include restoration, reconstruction, segmentation, edge and line detection or measurement of optical flow. These applications can be subdivided into ‘low level processing’ and ‘high level processing’ based on more abstract labels (like objects in the image). The following discussion mainly concentrates on image restoration.

4.3.1 Modeling an image

As a result of the properties found in Chapter 3, which showed that two adjacent pixels are strongly correlated, we expect that a Markov Field model based on small neighbourhoods for an image can already be very effective. Due to the Hammersley-Clifford theorem, this can be done by defining potentials of a Gibbs distribution, where the interactions between adjacent pixels contribute to the energy function. The resulting probability distribution $P(\mathbf{s})$ is called the *prior* model or the *a priori* information of the source image. From a qualitative point of view, the energy in the Gibbs distribution should favour configurations where neighbouring pixels have similar or identical states, that is, the energy E should contain terms like $const. \times |s_i - s'_i|$ where s'_i is the value of a neighbouring pixel of s_i. This will describe the ‘smoothness’ of the image, which can also be seen as a ferromagnetic term favouring adjacent particles in the same state. The constant is a parameter depending on the image, and has to be chosen so that the prior Gibbs distribution resembles the underlying image statistics as closely as possible.

The simplest model for the prior energy is a function that simulates a constant surface

$$E_{global,cs}(\mathbf{s}) = \sum_i K\delta(s_i - s_{\mathcal{N}_i}), \tag{4.12}$$

where $\delta(\cdot)$ is the Kronecker delta. Another form, providing a continuous function and no constant surface is the squared coupling

$$E_{global,sq}(\mathbf{s}) = \sum_i K(s_i - s_{\mathcal{N}_i})^2. \tag{4.13}$$

However, there is a problem in that discontinuities like lines or edges in the image are not allowed, since the energy would be very high and these features would vanish in the restoration process that minimizes the energy. A way out of this is to cut the function when it exceeds a given threshold and leave the energy constant for higher values. This can be achieved by the 'line process' introduced in [GG84].

Another formalism used for image restoration, and indeed similar to Markov Fields, which interprets the energy as a smoothness term and the noise as a closeness term, is *adaptive regularization* [Li95a, Li95b, RM90, PTK85]. The image in this case need not necessarily be defined on a discrete grid with a predefined set of states, but can be a two-dimensional function in $\mathbb{R}$. Let us assume $\mathbf{s}$ and $\mathbf{r}$ are functions of $x \in \mathbb{R}$ (in the range of x_{min} and x_{max}. Then

$$E_{global}(\mathbf{s}|\mathbf{r}) = V(\mathbf{s}|\mathbf{r}) + V(\mathbf{s}), \tag{4.14}$$

where

$$V(\mathbf{s}|\mathbf{r}) = \int_{x_{min}}^{x_{max}} \chi(x)\,[s(x) - r(x)]^2 \tag{4.15}$$

is the distance between restored and received data and $\chi(x)$ is a weighting function (corresponding to the error model). The term

$$V(\mathbf{s}) = \sum_{j=1}^{k} \lambda_j \int_{x_{min}}^{x_{max}} f\left(s(x)^{(j)}\right) \tag{4.16}$$

is the smoothness of the solution, with $s(x)^{(j)}$ being the j-th derivative of $s(x)$ and $f(\cdot)$ the cost function, which penalizes the discontinuities of the function $s(x)$ using the constant λ_j.

In general it is paramount to find a suitable form of the prior, since the qualitity of the restoration depends strongly on the correct parameters. If the prior is well-matched this will secure an optimal decoding process (see [PB95]). The power of the models introduced in this thesis lies in the fact, that their prior-energy have a free form which can adapt itself to the source via the paramters.

4.4 Restoring images

The ultimate goal of restoration and reconstruction is to recover the original image from a noisy candidate or from data which is only partially available. In other words:given the data (and some knowledge about the original) we want a restored image which is as close as possible to the original one. Closeness requires a measure of the error between two images. The knowledge about the original image is incorporated in a model, which is in our case an MRF, described by the corresponding neighbourhood system. During a noisy transmission the values of the pixels are changed randomly, which breaks up the neighbourhood relation. This leads to a change in the local potentials, and the received image **r** no longer conforms to the *prior* model. The transmission process also requires a model and is commonly described by the conditional probability $P(\mathbf{r}|\mathbf{s})$, which is the probability of receiving **r** if the original data was **s**. In the restoration procedure this process is inverted by maximizing the *a posterior* probability $P(\hat{\mathbf{s}}|\mathbf{r})$ or minimizing a predefined cost function. $\hat{\mathbf{s}}$ in this notation is the restored image and $P(\hat{\mathbf{s}}|\mathbf{r})$ the conditional probability for $\hat{\mathbf{s}}$ given **r**. Now since the noise is also described by a probability, the Bayes Equation (2.10) can be used

$$P(\hat{\mathbf{s}}|\mathbf{r}) = \frac{P(\mathbf{r}|\hat{\mathbf{s}})P(\hat{\mathbf{s}})}{P(\mathbf{r})}. \tag{4.17}$$

All the probabilities involved in the process can be written down as a Gibbs distribution of an exponential form, combining the prior and the noise in an energy function. The received image can then be interpreted as an external field coupled to the original. The strength of the coupling corresponds to the amount of noise; the lower the noise the stronger the coupling. This results in the total energy

$$E_{total}(\mathbf{s}|\mathbf{r}) = E_{prior}(\mathbf{s}) + E_{noise}(\mathbf{s}|\mathbf{r}), \tag{4.18}$$

which will be minimized during restoration. The last step is to find a suitable noise measure, which helps us to find the optimal $\hat{\mathbf{s}}$. From a Bayesian point of view, we want to find the estimator which minimizes the Bayes risk R (see Equation (2.11)).

4.4.1 Maximum a posterior (MAP)

The cost function of the MAP estimator is defined by

$$C(\mathbf{s}, \hat{\mathbf{s}}) = 1 - \delta(\mathbf{s}, \hat{\mathbf{s}}), \tag{4.19}$$

where $\delta(\mathbf{s}, \hat{\mathbf{s}}) = 1$ if and only if $\mathbf{s}$ is equal to $\hat{\mathbf{s}}$, which has the same cost for all configurations different from the original. The MAP estimator is then given by

$$\hat{\mathbf{s}}^{MAP} = \arg\ \max_{\mathbf{s}\in\mathbf{S}} P(\mathbf{s}|\mathbf{r}). \tag{4.20}$$

4.4.2 Marginal a posteriori mode (MPM)

The cost function of the MPM is defined as

$$C(\mathbf{s}, \hat{\mathbf{s}}) = \sum_{i\in\mathcal{G}} (1 - \delta(s_i, \hat{s}_i)), \tag{4.21}$$

where the sum runs over all sites of the grid. The solution for the best estimator is

$$\hat{s}_i^{MPM} = \arg\ \max_{s_i\in S_i} P(s_i|\mathbf{r}) \quad \forall i \in \mathcal{G}. \tag{4.22}$$

This corresponds to finding the exact value for each site separately.

4.4.3 Thresholded posterior mean (TPM)

Here, the cost function is:

$$C(\mathbf{s}, \hat{\mathbf{s}}) = \sum_{i\in\mathcal{G}} (s_i - \hat{s}_i)^2, \tag{4.23}$$

which gives the estimator

$$\hat{s}_i^{TPM} = \sum_{\mathbf{s}\in\mathbf{S}} s_i P(s_i|\mathbf{r}) \quad \forall i \in \mathcal{G}. \tag{4.24}$$

This estimator is used for image processing purposes, since it resembles the subjective error better than the other two. The eye is not sensitive to small differences in pixel values and does not require an exact match of the restored and original [MMP87, Li95a].

The decoding process, which is in this case the restoration of the image, is nothing else but the minimization of the Bayes risk and hence the finding of the estimators. Since the prior as well as the resulting a posteriori probabilities are fully described by a Gibbs distribution, and by the energy term, this amounts to finding the minimum energy of the whole system a common task in the simulation of physical systems. An additional advantage is that this representation makes it possible to investigate the model and its behavior from a physical viewpoint [PB95].

4.4.4 Coming to Monte Carlo

The minimization of the energy is a combinatorial optimization problem and special algorithms are required for this purpose. The most widely used algorithms are based on Monte Carlo methods which means stochastical optimization, for example the Metropolis algorithm [MRR$^+$53, PTVF92] (which should actually be the Metropolis-Rosenbluth-Rosenbluth-Teller-Teller algorithm to appreciate all authors properly) or simulated annealing [KGV83]. Other methods like genetic algorithms [Gol89, Col99] have also found large fields of applications [WCVG99].

The following concentrates on the standard Metropolis algorithm, which gives good results for the image restoration problem. However, the drawback is that for high levels of noise, many iterations are necessary, which leads to exececssive computational time.

In the general case, the Metropolis algorithm minimizes a function $E(\mathbf{s})$ with respect to $\mathbf{s}$ by an iterative procedure. During each step of the iteration the next configuration $\mathbf{s}'$ is tested against the function value of the current state $\mathbf{s}$ and is chosen if $\Delta E = E(\mathbf{s}') - E(\mathbf{s}) \leq 0$, that is the energy is lower. However, if $\Delta E > 0$, then there is a small probability $P = \exp(-\Delta E/T)$ that the the new state will be $\mathbf{s}'$ despite the fact that this gives a rise in energy. This can be achieved by comparing a random number – drawn from a uniform distribution in the interval $[0, 1)$ – with P. After a number of iterations the system will reach a local equilibrium which has minimum energy. The temperature T in our case is fixed and is implicitly contained in the parameters for the pixel couplings. Note that an extended version of the Metropolis algorithm is used in this thesis. This extension is described in Section 5.5.

Chapter 5

New physical models for images

The main requirement for the description of complex data like images is to find a simple form of the neighbourhood relation between the variables involved, which is locally defined, yet models the statistical properties of the whole system as closely as possible. This simple model should be physically tractable and contain only a few parameters, and thereby be easy to estimate from the data. The general framework of the Gibbs Random Fields meets both of these requirements and the formalism based on energy terms and canonical distributions provides a good analogy to the physical world.

5.1 Physical background

Starting from the physical side of the image processing problem, we write the canonical partitition function for a statistical system as

$$Z = \sum_{\{\mathbf{s}\}} \mathrm{e}^{-\beta \mathcal{H}(\mathbf{s})}, \tag{5.1}$$

where $\{\mathbf{s}\}$ is the set off all possible configurations. Consider the set of labels s_i defined on the N points (vertices) of a lattice, such that the Hamiltonian is a function of the form

$$\begin{aligned} \mathcal{H}(\mathbf{s}) &= \sum_{\langle ij \rangle} E(s_i, s_j) \ \text{(pairwise)} && (5.2) \\ &= \sum_i \sum_{j \in \mathcal{N}_i} V_{\mathcal{N}}(s_i, s_j) \ \text{(nearest-neighbours)}, && (5.3) \end{aligned}$$

where $\langle ij \rangle$ denotes the sum over all possible neighbours, and after defining a neighbourhood system $\mathcal{N}_i$ (for the vertex i) j is one of the nearest-neighbours. The local potential $V_{\mathcal{N}}$ is often called the *clique-potential* as it is defined for the corresponding neighbourhood. This follows the notation introduced in Section 4.2.2. The corresponding Boltzmann-distribution is then

$$P(\mathbf{s}) = \frac{1}{Z} \mathrm{e}^{-\beta \mathcal{H}(\mathbf{s})}. \tag{5.4}$$

However, the difference between the physical world and image processing is that in the first case the potentials, the parameters and the energy are known and possible states as well as thermodynamical quantities can be calculated, whereas in the second case a single configuration (the whole image) is known, but not the parameters. However, thermodynamical quantities can be *measured* in the image and after a model has been found, the parameters can be estimated from the data by requesting that calculated quantities approximate the measured quantities. Examples are the mean energy

$$\langle E \rangle = \frac{1}{Z} \sum_{\{\mathbf{s}\}} \mathcal{H}(\mathbf{s}) \mathrm{e}^{-\beta \mathcal{H}(\mathbf{s})} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta}, \tag{5.5}$$

the entropy

$$H_{phy} = k(\ln Z + \beta \langle E \rangle) \tag{5.6}$$

as a measure of information for the system, and especially correlations

$$\langle s_i s_j \rangle = \frac{1}{Z} \sum_{\{\mathbf{s}\}} \sum_{i,j} s_i s_j \mathrm{e}^{-\beta \mathcal{H}(\mathbf{s})} = -\frac{1}{Z} \frac{1}{\beta} \frac{\partial Z}{\partial J}, \tag{5.7}$$

where J is the parameter corresponding to the correlation (coupling constant including β).

5.1.1 Ising Model – a one-dimensional information system

To illustrate the model building approach the simplest case of the one-dimensional Ising model without an external field is shown here. Consider a chain with N positions. At each point there is a spin which is either up or down. Two neighbouring spins are coupled by a coupling constant J. The Hamiltonian of the Ising model is

$$\mathcal{H} = J \sum_{i=1}^{N} s_i s_{i+1} \text{ where } s_i \in \{-1, +1\}. \tag{5.8}$$

The partition function (with $K = -\beta J$) is

$$Z = \sum_{\{\mathbf{s}\}} \prod_{i=1}^{N} \mathrm{e}^{K s_i s_{i+1}} \tag{5.9}$$

and was solved first by Ising using the transfer-matrix method (see also [Dom74, Kog79]). This matrix is defined by the four possible states of the two spins in Equation (5.9)

$$\bar{\mathbf{T}}(s_i, s_{i+1}) = \begin{pmatrix} \mathrm{e}^{K} & \mathrm{e}^{-K} \\ \mathrm{e}^{-K} & \mathrm{e}^{K} \end{pmatrix}. \tag{5.10}$$

Now Z can be rewritten as

$$\begin{aligned} Z &= \sum_{\{\mathbf{s}\}} \prod_{i=1}^{N} \bar{\mathbf{T}}(s_i, s_{i+1}) \qquad (5.11) \\ &= \mathit{trace}\ \bar{\mathbf{T}}(s_i, s_{i+1})^N \\ &= \sum_i \lambda_i^N, \end{aligned}$$

where we assume a chain which has closed boundaries. The eigenvalues of the transfer matrix are

$$\lambda_{\pm} = \mathrm{e}^{K} \pm \mathrm{e}^{-K} \tag{5.12}$$

and the partition function

$$Z = 2^N \left(\cosh^N K + \sinh^N K\right). \tag{5.13}$$

For a large system in the thermodynamical limit where $N \to \infty$ this simplifies to

$$Z = \lambda_0^N \left(1 + \left(\frac{\lambda_1}{\lambda_0}\right)^N\right) \overset{N\to\infty}{\longrightarrow} \lambda_0^N. \tag{5.14}$$

This makes it easy to calculate the thermodynamical quantities of the system, for example the entropy (per particle)

$$H_{phy} = k\left(\ln\left(2\cosh K\right) - K \tanh K\right) \tag{5.15}$$

or nearest-neighbour correlations (from Equation (5.7))

$$\begin{aligned}
\langle s_i s_{i+1} \rangle &= \frac{1}{N}\frac{\partial}{\partial K}\ln Z \qquad (5.16)\\
&= \frac{1}{N}\frac{\partial}{\partial K}\ln\left(2^N \cosh^N K\right)\\
&= \frac{\sinh K}{\cosh K}\\
&= \tanh K.
\end{aligned}$$

This can be solved for K as

$$K = \tanh^{-1}(\langle s_i s_{i+1} \rangle). \qquad (5.17)$$

Now starting from the image processing side we can represent a black and white image row by an Ising model, where (-1) stands for black and (+1) for white pixels. The correlations between two neighbouring pixels can be measured from an image by running along the row and calculating

$$\begin{aligned}
\langle s_i s_{i+1} \rangle_{data} &= \frac{1}{N}\sum_{i=1}^{N} s_i s_{i+1} \qquad (5.18)\\
&= \sum_{\Delta s_i=-1}^{+1} p(\Delta s_i)\Delta s_i,
\end{aligned}$$

where Δs_i is $+1$ for $s_i = s_{i+1}$ or -1 else. $p(\Delta s_i)$ is the probability for the joined state Δs_i of the two pixels.

If the image data is a valid member of the model that was assumed, then the value of the measured correlation has to be identical to the one predicted by the model. This means we demand the equality

$$\langle s_i s_{i+1} \rangle \overset{!}{=} \langle s_i s_{i+1} \rangle_{data} \qquad (5.19)$$

and we can simply calculate K by inserting this result in Equation (5.17). K contains the coupling constant J and the temperature T. Note also that this holds for systems where the neighbourhood is much smaller than the size of the whole system. With the entropy of the system readily given by Equation (5.15) it is also possible to find the entropy as defined in information theory. However the parameter k has to be changed, since the definitions of the two entropies are different. In the physical case k is simply the

Boltzmann constant. For H_{inf} we can derive the constant by looking at the simple system where all spins are uncorrelated and the temperature goes to infinity. In this case $\langle s_i s_{i+1} \rangle = 0$ and hence $K = 0$. The partition function then simply sums all identical configurations and $Z = 2^N$. The information carried by one spin in the system is then $H_{inf} = 1$ bit (per particle). On the other hand $H_{phy} = k \ln 2$ and this gives $k_{inf} = 1/ln2$. This answers the question: 'How many floppy discs do I need to store my Ising system ?'

5.2 Finding the coupling constants of an image

As has been shown, finding the GRF for any given image is an inverse problem from the physical point of view. While the coupling constants are known in the later case, for an image they need to be found. After having found a suitable model, the correlations $\langle image \rangle$ can be measured in the image. Now if this is indeed a physical system, the correlations have to be identical to the ones predicted by the theory, $\langle theory \rangle$. In the simple case of the one-dimensional Ising model this is easy, since the only constant K can be derived analytically from the correlations $\langle s_i s_{i+1} \rangle$. However, for the two-dimensional system the equality $\langle image \rangle = \langle theory \rangle$ has to be reached by numerical methods. The general requirement is that the equality

$$\langle m \rangle_{theory} = \langle m \rangle_{image} \tag{5.20}$$

holds for all possible correlations m. For the two-dimensional case this can be done using a variational approach. Alternatively one can try to approximate the two-dimensional model itself by simpler models. As will be shown this is actually a very efficient technique for image processing purposes.

5.3 Two-dimensional models

The transfer-matrix method introduced in the last section can be used to solve the two-dimensional Ising model. In this case $\bar{\mathbf{T}}$ in Equation (5.11) no longer connects two nearest-neighbour lattice points, but two one-dimensional chains (rows) of the grid. This leads to the general expression ($\mathcal{H}$ will now contain the constant β)

$$\begin{aligned} Z &= \sum_{\{\mathbf{s}\}} \prod_{\langle ij \rangle} \mathrm{e}^{\mathcal{H}(s_i, s_j)} \\ &= \sum_{\{\vec{s_1}\}} \sum_{\{\vec{s_2}\}} \cdots \sum_{\{\vec{s}_{N_y}\}} \prod_{i=1}^{N_y} \mathrm{e}^{\mathcal{H}(\vec{s_i}, \vec{s}_{i+1})}, \end{aligned} \tag{5.21}$$

where this reflects a model with nearest neighbours. Note that the Hamiltonian depends on the two rows $\vec{s_i}, \vec{s}_{i+1}$ and the sum is over all configurations of these two. Now from equation (5.11) we know that the eigenvalues of the transfer matrix have to be found and this gives the generalized eigenvalue problem

$$\int \mathrm{d}\vec{s} \psi_\alpha(\vec{s}) \exp\left(\mathcal{H}(\vec{s}, \vec{s}\,')\right) = \lambda_\alpha \psi_\alpha(\vec{s}\,'), \tag{5.22}$$

with a summation (integration) over all states $\vec{s}$, where the transfer matrix can be written as

$$\bar{\mathbf{T}}(\cdot) = \int \mathrm{d}\vec{s} \cdot \exp\left(\mathcal{H}(\vec{s}, \vec{s}\,')\right) \tag{5.23}$$

to give a more compact form

$$\bar{\mathbf{T}} |\alpha\rangle = \lambda_\alpha |\alpha\rangle. \tag{5.24}$$

In this case $|\alpha\rangle$ is an eigenvector of $\bar{\mathbf{T}}$ representing one row and λ_α the corresponding eigenvalue. The following assumes that $\bar{\mathbf{T}}$ is symmetric. Written in the base of the normalized eigenvectors the matrix elements can be determined by

$$\bar{\mathbf{T}}_{\alpha,\beta} = \langle \alpha | \bar{\mathbf{T}} | \beta \rangle = \int \mathrm{d}\vec{s} \int \mathrm{d}\vec{s}\,' \langle \alpha(\vec{s}) | \exp\left(\mathcal{H}(\vec{s}, \vec{s}\,')\right) | \beta(\vec{s}\,') \rangle. \tag{5.25}$$

For large systems – in analogy to the one-dimensional example – the partition function simplifies to

$$\begin{aligned} Z_{2D} &= Tr\ \bar{\mathbf{T}}^{N_y} \\ &= \sum_i \lambda_i^{N_y} \\ &= \lambda_0^{N_y} \left(1 + \left(\frac{\lambda_0}{\lambda_1} \right)^{N_y} + \ldots \right) \xrightarrow{N_y \to \infty} \lambda_0^{N_y}, \end{aligned} \tag{5.26}$$

where the eigenvalues λ_i can be ordered as $\lambda_0 > \lambda_1 > \lambda_2 ...$. This thermodynamical limit allows one to write the nearest-neighbour correlations of the system as

$$\begin{aligned} \langle\langle \bar{\mathbf{A}}(\vec{s_i}, \vec{s_{i+1}}) \rangle\rangle &= \frac{\partial}{\partial \alpha} \ln Z \\ &= \frac{1}{Tr(\bar{\mathbf{T}})} Tr(\bar{\mathbf{A}}\bar{\mathbf{T}}) \\ &= \frac{1}{\lambda_0} \langle 0|\bar{\mathbf{A}}\bar{\mathbf{T}}|0\rangle \end{aligned} \tag{5.27}$$

where the brackets denote the correlation and the average over the ensemble. α is the parameter corresponding to the correlation $\bar{\mathbf{A}}$. Since the MRFs require local properties only, it suffices to determine the largest eigenvalue and eigenvector of $\bar{\mathbf{T}}$ to find these correlation functions.

To summarize this: after defining the model and the Hamiltonian the largest eigenvalue and eigenvector of the transfer matrix have to be found.

5.3.1 Solution to the transfer operator eigenvalue problem

A well known approximation to the solution of the eigenvalue problem

$$\bar{\mathbf{T}}|\alpha\rangle = \lambda|\alpha\rangle \tag{5.28}$$

is the Rayleigh-Ritz variational approach. It is used in quantum mechanics to determine the ground state of a given Hamiltonian and approximate the largest eigenvalue λ_0 of $\bar{\mathbf{T}}$, where the approximation $\tilde{\lambda}_0 \leq \lambda_0$.

If the eigenvector of the ground state is already known, the true eigenvalue λ_0 can be found using the expectation value

$$\lambda_0 = \frac{\langle 0|\bar{\mathbf{T}}|0\rangle}{\langle 0|0\rangle} \tag{5.29}$$

Generally we can choose a model in which the transfer operator can be represent in a base where all matrix elements are non-negative and the eigenvector corresponding to the largest eigenvalue can be written in exponential form (see [Ruj79])

$$|0\rangle = \exp\left(|\phi_0\rangle\right). \tag{5.30}$$

Inserting this into Equation (5.29) gives

$$\begin{aligned}
\lambda_0 &= \frac{\langle 0|\bar{\mathbf{T}}|0\rangle}{\langle 0|0\rangle} \\
&= \int \mathrm{d}\vec{s} \int \mathrm{d}\vec{s}\,' \exp\left(\phi_0(\vec{s}) + \mathcal{H}(\vec{s},\vec{s}\,') + \phi_0(\vec{s}\,')\right) / \int \mathrm{d}\vec{s} \exp\left(2\phi_0\vec{s}\right) \\
&= Z^{(n)}/Z^{(d)}. \qquad (5.31)
\end{aligned}$$

This is the ratio of two partition functions. The numerator (denoted by index n) has the Hamiltonian

$$\mathcal{H}^{(n)} = \phi_0(\vec{s}) + \mathcal{H}(\vec{s},\vec{s}\,') + \phi_0(\vec{s}\,'), \qquad (5.32)$$

which is a system consisting of two rows, and the denominator (denoted by index d)

$$\mathcal{H}^{(d)} = 2\phi_0(\vec{s}), \qquad (5.33)$$

which is the partition function of a one-dimensional chain.

The correlations for the GRF models can then be derived using Equation (5.27)

$$\langle\langle \bar{\mathbf{A}}(\vec{s_i}, \vec{s_{i+1}})\rangle\rangle = \frac{1}{\lambda_0^{(n)}} \frac{\langle \phi_0^{(n)} | \bar{\mathbf{A}} \bar{\mathbf{T}}^{(n)} | \phi_0^{(n)} \rangle}{\langle \phi_0^{(n)} | \phi_0^{(n)} \rangle}, \qquad (5.34)$$

where $\lambda_0^{(n)}$, $|\phi_0^{(n)}\rangle$ and $\bar{\mathbf{T}}^{(n)}$ are the eigenvalue, eigenvector and transfer matrix of the numerator, respectively.

The next step is to find an appropriate trial function $|\{a\}^{tf}\rangle$, which can be used to maximize the Rayleigh ratio via a set of parameters. This then results in

$$\lambda_{\{a\}}^{tf} = \frac{\langle \{a\}^{tf} | \bar{\mathbf{T}} | \{a\}^{tf} \rangle}{\langle \{a\}^{tf} | \{a\}^{tf} \rangle} \leq \lambda_0 \qquad (5.35)$$

as an approximation to the largest eigenvalue.

The remaining problem is now the selection of the trial function, which should be guided by the properties of the operator. For the Ising model a promising approach is the sum of products of spin variables

$$\phi_0 = a_0 \sum_i s_i + a_{01} \sum_i s_i s_{i+1} + a_{02} \sum_i s_i s_{i+2} + a_{012} \sum_i s_i s_{i+1} s_{i+2} + \ldots \qquad (5.36)$$

Since the ratio Equation (5.35) is a lower approximation to the eigenvalue of the ground state, $\lambda_{\{a\}}$ has to be maximized with respect to the parameters $\{a\}$. The necessary condition for this is

$$\frac{\partial}{\partial a_i}\lambda_{\{a_i\}} = 0 \tag{5.37}$$

for all a_i. After applying this to Equation (5.35), we get

$$\langle\langle \prod_{i\in\{a_j\}} s_i\rangle\rangle^{(n)} = \langle\langle \prod_{i\in\{a_j\}} s_i\rangle\rangle^{(d)} \tag{5.38}$$

for all j, where the notation $i \in \{a_j\}$ indicates, that the product of the spins corresponding to the parameter a_j are to be used. The brackets stand for the expectation values of the observable numerator (n) and denominator (d) and can be interpreted as the correlations associated with the parameter.

5.3.2 Two-dimensional Ising model for black and white images

The Ising model provides a good start for images having only two possible pixel values:black and white. These can be represented by the states $s_{black} = -1$ and $s_{white} = +1$. The GRF has the Hamiltonian

$$\mathcal{H} = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(K_x s_i^j s_{i+1}^j + K_y s_i^j s_i^{j+1}\right), \tag{5.39}$$

where K_x and K_y are the coupling constants in x- and y-direction.
The correlations corresponding to K_x and K_y can be measured in the image

$$\langle ss\rangle_x = \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} s_i^j s_{i+1}^j \tag{5.40}$$

$$\langle ss\rangle_y = \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} s_i^j s_i^{j+1}, \tag{5.41}$$

where s_i^j is the value of the pixel state at the lattice vertex (i, j).
The Hamiltonian of two rows $\vec{s}, \vec{s}\,'$ is

$$\mathcal{H}(\vec{s}, \vec{s}\,') = \frac{K_y}{2}\left(\sum_i s_i s_i' + \sum_i s_{i+1}s_{i+1}'\right) + \frac{K_x}{2}\left(\sum_i s_i s_{i+1} + \sum_i s_i' s_{i+1}'\right). \tag{5.42}$$

As mentioned above, the trial function for the variational approach can be selected on the base of spin products Equation (5.36). However, the approximation will only contain the first two terms of the sum

$$\psi^{tf} = \mathrm{e}^{a_0 \sum_i s_i + a_{01} \sum_i s_i s_{i+1}}, \tag{5.43}$$

which results in the ratio (from Equation (5.31))

$$\tilde{\lambda}_0 = \tilde{\lambda}_0^{(n)} / \tilde{\lambda}_0^{(d)}, \tag{5.44}$$

where

$$\tilde{\lambda}_0^{(n)} = \sum_{\{\vec{s}\}} \sum_{\{\vec{s}\,'\}} \mathrm{e}^{a_0 \left(\sum_i s_i + \sum_i s'_i\right) + \left(\frac{K_x}{2} + a_{01}\right)\left(\sum_i s_i s_{i+1} + \sum_i s'_i s'_{i+1}\right) + \frac{K_y}{2}\left(\sum_i s_i s'_i + \sum_i s_{i+1} s'_{i+1}\right)} \tag{5.45}$$

for the numerator and

$$\tilde{\lambda}_0^{(d)} = \sum_{\{\vec{s}\}} \mathrm{e}^{2a_0 \sum_i s_i + 2a_{01} \sum_i s_i s_{i+1}} \tag{5.46}$$

for the denominator.The transfer matrix of the numerator consists in this case of 4x4 elements. The denominator is simply the one-dimensional case, albeit with an external field $2a_0$. To illustrate this, the numerator operator $\bar{\mathbf{T}}^{(n)}$ may be represented by

$$\begin{array}{c|cccc} {}_{s_{i+1}s'_{i+1}} \, {}^{s_i s'_i} & -- & -+ & +- & ++ \\ \hline -- & \omega(0) & \omega(1) & \omega(1) & \omega(2) \\ -+ & \omega(1) & \omega(3) & \omega(5) & \omega(4) \\ +- & \omega(1) & \omega(5) & \omega(3) & \omega(4) \\ ++ & \omega(2) & \omega(4) & \omega(4) & \omega(6) \end{array}, \tag{5.47}$$

with the elements

$$\begin{aligned} \omega(0) &= \mathrm{e}^{-2a_0 + 2a_{01} + K_x + K_y} \\ \omega(1) &= \mathrm{e}^{-a_0} \\ \omega(2) &= \mathrm{e}^{-2a_{01} - K_x + K_y} \\ \omega(3) &= \mathrm{e}^{+2a_{01} + K_x - K_y} \\ \omega(4) &= \mathrm{e}^{a_0} \\ \omega(5) &= \mathrm{e}^{-2a_{01} - K_x - K_y} \\ \omega(6) &= \mathrm{e}^{2a_0 + 2a_{01} + K_x + K_y}. \end{aligned}$$

The denominator $\bar{\mathbf{T}}^{(d)}$ reads

$$\begin{array}{cc} s_{i+1} \quad {}^{s_i} & - \qquad + \end{array} \tag{5.48}$$

$$\begin{array}{c} - \\ + \end{array} \begin{pmatrix} \omega(7) & \omega(8) \\ \omega(8) & \omega(9) \end{pmatrix}, \tag{5.49}$$

with elements

$$\begin{aligned} \omega(7) &= \mathrm{e}^{-2a_0+2a_{01}} \\ \omega(8) &= \mathrm{e}^{-2a_{01}} \\ \omega(9) &= \mathrm{e}^{2a_0+2a_{01}}. \end{aligned}$$

The numerical procedure consists of first measuring the correlations in the image, $\langle image \rangle$ (Equation (5.40)). Then, after initializing K_x, K_y, a_0 and a_{01} the Rayleigh ratio is determined from $\tilde{\lambda}_0^{(n)}$ and $\tilde{\lambda}_0^{(d)}$ and the corresponding correlations $\langle (n) \rangle$,$\langle (d) \rangle$ are used to find the gradient for the maximization. When the maximum has been found the correlations are compared to the measured ones and K_x and K_y are altered until these correlations are equal.

5.3.3 Two-dimensional Z_q model for grey level images

The next step towards a more general description of natural images is to extend the range of allowed states to the range of possible brightness levels of a pixel (rather than just black and white). The number of grey levels denoted is usually 256, corresponding to 8 bit per pixel.

There exist several models of physical systems which describe the interaction between points on the lattice, each one having more than two possible states. These are the Potts, Vector-Potts, XY and the Z_q model [Dom74, RWFF81, DPR81].

From these models the Z_q model is chosen for the image processing task, since the parameters required in the Hamiltonian can be calculated from the data and the energy has a form which allows an easy interpretation of the correlations involved. One disadvantage compared to other models is that the number of coupling constants is equal to the number of brightness levels. Usually the number of parameters should be as small as possible, but this number is still very small compared to the storage requirements of an image.

The Z_q model is a cyclic model containing terms that require the angle between two states – 'pointing' in a certain direction. Mapping the brightness levels directly to the states would mean that on one hand the black and white states are direct neighbours when calculating the angle counterclockwise, but on the other hand separated by all remaining states when going clockwise. To circumvent this, the g levels of brightness are extended to $q = 2g - 2$ states of the Z_q model, covering the whole angle of 2π. The mapping is displayed in Figure 5.1. Note that this is more a theoretical problem, the actual calculations do not contain any of the extended states directly.

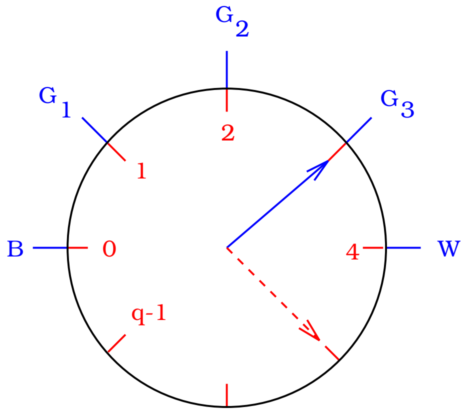


Figure 5.1: Mapping the grey levels (g) of an image onto q states of the Z_q model. **B** is the black pixel and **W** the white pixel value. $\mathbf{G}_n$ are intermediate levels of brightness.

In order to simplify the notation, we assume that the coupling constants in the x and y directions are equal. Furthermore, we use the variable l_x^y to denote a state at (x, y) in the Z_q model in order to distinguish it from the two states s_i of the Ising model. The Hamiltonian of the Z_q model is:

$$\mathcal{H}^{(Z(q))} = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \sum_{m=1}^{q/2} K_m \left[\cos\left(\frac{2\pi}{q} m (l_x^y - l_{x+1}^y) \right) + \cos\left(\frac{2\pi}{q} m (l_x^y - l_x^{y+1}) \right) \right]. \tag{5.50}$$

This notation illustrates the adaptive property of the Z_q model: the couplings corresponding to the energy are written in the base of cosine terms and can be seen as a kind of discrete cosine transformation. Hence, it is not necessary to search for the coupling **function** which fits the data, the only task is to determine the **parameters** K_m. The model itself is invariant under exchange of the two neighbouring points, which is a result of the symmetry of the cosine function. This means that we do not assume any global spatial gradients of the brightness levels in the image.

To simplify the notation we define

$$c(m,\tilde{l}) := \cos(\frac{2\pi}{q}m\tilde{l}) \tag{5.51}$$

for the model with q levels.

In analogy to the Ising model, the nearest-neighbour correlations can be calculated from the image and since there are $q/2$ coupling constants, we also have $q/2$ correlations

$$\langle m \rangle = \frac{1}{N_x}\frac{1}{N_y}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\left(c(m,l_x^y - l_{x+1}^y) + c(m,l_x^y - l_x^{y+1})\right), \tag{5.52}$$

where m runs from 1 to $q/2$ and l_x^y is the brightness value of the pixel at coordinates (x,y). The Hamiltonian between two rows is

$$\begin{aligned} \mathcal{H}(\vec{l},\vec{l}') \;=\; \textstyle\sum_i \sum_{m=1}^{q/2} & \tfrac{K_m}{2}\left(c(m,l_i - l_{i+1}) + c(m,l'_i - l'_{i+1})\right) \\ + & \tfrac{K_m}{2}\left(c(m,l_i - l'_i) + c(m,l_{i+1} - l'_{i+1})\right). \end{aligned} \tag{5.53}$$

As stated in the model with two states, the trial function in the variational ansatz can be written in the base of the states, but this time the term $s_i s'_i$ corresponds to the cosine function $\cos(\frac{2\pi}{q}m(l_i - l'_i))$. The trial function is

$$\psi^{tf} = \exp\sum_{m=1}^{q/2}\left(a_0^{(m)}\sum_i \cos\left(\frac{2\pi}{q}ml_i\right) + a_{01}^{(m)}\sum_i \cos\left(\frac{2\pi}{q}m(l_i - l_{l+1})\right)\right). \tag{5.54}$$

The numerator and denominator Hamiltonians in the Rayleigh ratio are

$$\begin{aligned} h^{(n)} = \textstyle\sum_{m=1}^{q/2} & \left(a_0^{(m)}/2\right)\left(c(m,l_i) + c(m,l'_i) + c(m,l_{i+1}) + c(m,l'_{i+1})\right) \\ + & \left(a_{01}^{(m)} + K_m/2\right)\left(c(m,l_i - l_{i+1} + c(m,l'_i - l'_{i+1})\right) \\ + & (K_m/2)\left(c(m,l_i - l'_i) + c(m,l_{i+1} - l'_{i+1})\right) \end{aligned} \tag{5.55}$$

and

$$h^{(d)} = \sum_{m=1}^{q/2} \left[a_0^{(m)} \left(c(m, l_i) + c(m, l_{i+1}) \right) + 2a_{01}^{(m)} c(m, l_i - l_{i+1}) \right], \qquad (5.56)$$

respectively.

The transfer matrix for the numerator and denominator now have the dimensions of $(q/2)^2 \times (q/2)^2$ and $(q/2) \times (q/2)$ respectively. To determine the constants K_m the same method as in the two state case is used (the Z_q model with two states it identical to the Ising model), however the computational effort is much higher, which is due to the fact that the transfer matrix now has many more elements.

5.4 One-dimensional models

One of the disadvantages of the two-dimensional solution via the variational approach is that the solution to the transfer matrix eigenvalue problem becomes numerically intractable for large q. The maximum value for the number of grey levels which gives a solution in a satisfactory (less than a few seconds) time is $g = 6$, corresponding to $q = 10$ spin states. The transfer matrix in Equation (5.31) contains 100×100 elements and one has to ensure the equality between measured and calculated correlations which requires the calculation of the eigenvectors and eigenvalues in the optimization procedure for the Rayleigh ratio, which is again part of the numerical minimization for Equation (5.20).

A solution to this problem is to approximate the two-dimensional system with a one-dimensional approach. This is known as high or low temperature expansion, where the two-dimensional partition function is expanded in a series which is then truncated after a few terms [Dom74, Kog79, Sav80]. As the results of the experiments with natural images suggest, it might already be sufficient to use the first term of the series only. After the expansion and using the first term, the resulting one-dimensional system can be solved analytically, hence avoiding the need for a numerical optimization procedure.

5.4.1 One-dimensional Ising model for black and white images

Although it is possible to find a solution to the two-dimensional Ising model using the variational approach due to the small number of states and small matrices, the one-dimensional approximation is given here as a motivation and to illustrate its properties.

The general Ising model (interaction between nearest neighbours) has the Hamiltonian (see Equation (5.8))

$$\mathcal{H} = -J \sum_{\langle ij \rangle} s_i s_j \tag{5.57}$$

and the partition function

$$Z = \sum_{\{\mathbf{s}\}} e^{-\beta J \sum_{\langle ij \rangle} s_i s_j}. \tag{5.58}$$

The first step in the duality transformation is to rewrite the sum

$$Z = \sum_{\{\mathbf{s}\}} \prod_{\langle ij \rangle} e^{K s_i s_j}, \tag{5.59}$$

where $K = -\beta J$. Using the fact that $s = \pm 1$ one gets

$$\mathrm{e}^{Ks} = \cosh K + s \sinh K. \tag{5.60}$$

This results in the partition function

$$\begin{aligned} Z &= \sum_{\{\mathbf{s}\}} \prod_{\langle ij \rangle} (\cosh K + s_i s_j \sinh K) \\ &= (\cosh K)^{N_x N_y} \sum_{\{\mathbf{s}\}} \prod_{\langle ij \rangle} (1 + s_i s_j \tanh K), \end{aligned} \tag{5.61}$$

where the sum of spin pairs in the exponent are now written as a product of pairs in the exponential/tanh function. Factorizing this product yields the following terms:

$$\begin{aligned} Z = (\cosh K)^{N_x N_y} \sum_{\{\mathbf{s}\}} \quad (1 \quad & + \sum_{\langle ij \rangle} v s_i s_j + \sum_{\langle ijkl \rangle} v^2 s_i s_j s_k s_l \\ & + \sum_{\langle ijklmnop \rangle} v^4 s_i s_j s_k s_l s_m s_n s_o s_p + \ldots), \end{aligned} \tag{5.62}$$

where $v = \tanh K$. However, since all the spin products which do not have an even exponent of the product $s_i s_i$ drop out after the summation over all states, this expansion simplifies to

$$Z = (\cosh K)^{N_x N_y} (2^{N_x N_y} + v^4 N_x N_y + \ldots), \tag{5.63}$$

where the second term (v^4) corresponds to a closed loop of four points $s_i s_j s_j s_k s_k s_l s_l s_i$.

For small K, the approximation $v = \tanh K \approx K$ holds and we drop the terms containing v^4. This is known as 'high-temperature expansion', since for high temperature $K \to 0$. Note that the same applies to low-temperatures, which is a duality transformation of the high T regime. The result

$$Z = (2\cosh(K))^{N_x N_y} \tag{5.64}$$

is identical to the partition function of the one-dimensional Ising chain.
By now using different coupling constants for the horizontal and vertical directions (rows and columns) it can readily be shown that the approximation gives a product of two independent partition functions,

$$Z^{(xy)} = (2\cosh K_x)^{N_x}(2\cosh K_y)^{N_y}. \tag{5.65}$$

The correlations can be calculated separately as

$$\begin{aligned} \langle s_x s_{x+1} \rangle &= \tanh K_x && (5.66) \\ \langle s_y s_{y+1} \rangle &= \tanh K_y. && (5.67) \end{aligned}$$

This enables us to treat the image as a system of **independent** rows and columns. An extension to higher orders results in a stronger interaction between the two. The error introduced by the truncated expansion can be described in a qualitative manner. At high temperatures the correlations in the model would be comparatively small. If the measurement in the image (which is independent of the expansion) yields large values this would correspond to a lower temperature than expected for the true two-dimensional system. This means the values of the couplings K are large and the interaction between two pixels stronger. The result is an 'over-smoothing' of the restored image.

5.4.2 One-dimensional Z_q model for grey level images

In analogy to the approximation of the Ising model using two one-dimensional systems, the same procedure can be applied to the Z_q model with q states. The partition function is (from Equation (5.50) and 5.51)

$$Z_{Zq} = \sum_{\{\mathbf{l}\}} \exp\left(\sum_{\langle ij \rangle} \sum_{m=1}^{q/2} K_m \left(c(m, l_i - l_j) \right) \right). \tag{5.68}$$

To simplify the notation we again assume the same couplings for x and y directions. Additionally, the definitions

$$\omega(n) := \mathrm{e}^{\sum_{m=1}^{q/2} K_m \cos\left(\frac{2\pi}{q} mn\right)} \tag{5.69}$$

and

$$\alpha := \frac{2\pi}{q} \tag{5.70}$$

are used.

After rewriting the exponential function as a product, the sum of the spin pairs in the exponents are now products of the Boltzmann functions

$$\begin{aligned} Z_{Zq} &= \sum_{\{\mathbf{l}\}} \prod_{\langle ij \rangle} \omega(l_i - l_j) \\ &= \sum_{\{\mathbf{l}\}} \prod_{\langle ij \rangle} \sum_{k=0}^{q-1} \delta_{k,|l_i-l_j|} \omega(k), \end{aligned} \tag{5.71}$$

where δ is the Kronecker delta.

Moving the factor $\omega(0)$ out of the sum gives

$$Z_{Zq} = \omega(0)^{N_x N_y} \sum_{\{\mathbf{l}\}} \prod_{\langle ij \rangle} \left(1 + \sum_{k=1}^{q-1} \delta_{k,|l_i-l_j|} \frac{\omega(k)}{\omega(0)} \right), \tag{5.72}$$

then multiplying the terms of the product (up to first order), rewriting the sum using the δ function and summing over all states we obtain

$$\begin{aligned} Z_{Zq} &= \omega(0)^{N_x N_y} \sum_{\{\mathbf{l}\}} \left(1 + \sum_{\langle ij \rangle} \sum_{k=1}^{q-1} \delta_{k,|l_i-l_j|} \frac{\omega(k)}{\omega(0)} + \ldots \right) \\ &\approx \omega(0)^{N_x N_y} \left(q^{N_x N_y} + \left(q \sum_{k=1}^{q-1} \frac{\omega(k)}{\omega(0)} \right)^{N_x N_y} \right), \end{aligned} \tag{5.73}$$

which is the largest eigenvalue of the transfer matrix of the one-dimensional Z_q model and hence the first order approximation.

The remaining problem is now the calculation of the $q \times q$ matrix of the transfer operator, the determination of the (largest) eigenvalue, its corresponding eigenvector and finally the analytical solution to the problem of finding the coupling constants K_m from the correlations.

From the previous results the partition function of the one-dimensional Z_q model is

$$Z = \sum_{\{\mathbf{l}\}} \prod_{i=1}^{N} \exp\left(\sum_{m=1}^{q/2} K_m \cos\left(\frac{2\pi}{q} m (l_i - l_{i+1}) \right) \right), \tag{5.74}$$

where $\mathbf{l}$ is one chain of points (with periodic boundaries). The elements of the transfer matrix are

$$T(l,l') = \omega(|l-l'|) = \exp\left(\sum_{m=1}^{q/2} K_m \cos(\alpha m (l-l')) \right) \tag{5.75}$$

and $\bar{\mathbf{T}}$ is cyclic due to the cosine terms

$$\bar{\mathbf{T}} = \begin{pmatrix} \omega(0) & \omega(1) & \omega(2) & \dots & \omega(2) & \omega(1) \\ \omega(1) & \omega(0) & \omega(1) & \dots & \omega(3) & \omega(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega(1) & \omega(2) & \omega(3) & \dots & \omega(1) & \omega(0) \end{pmatrix}. \tag{5.76}$$

Now $\bar{\mathbf{T}}$ can be written using the so called ‘elementary cyclic’ matrix

$$\bar{\mathbf{T}} = \omega(0)\bar{\mathbf{E}} + \sum_{m=1}^{q/2-1} \omega(m) \left(\bar{\mathbf{\Omega}}^m + (\bar{\mathbf{\Omega}}^m)^{-1} \right) + \omega(q/2)\bar{\mathbf{\Omega}}^{q/2}, \tag{5.77}$$

where the last term in the sum appears only if q is even. $\bar{\mathbf{E}}$ is the unit matrix. The matrix elements of the quadratic $q \times q$ matrix Ω are defined as

$$\Omega(i,j) = \delta_{i+1,j} + \delta_{i,q-1}\delta_{j,1} \tag{5.78}$$

and have the characteristic that all elements of the upper sub-diagonal and the element in the lower left corner are 1. Multiplying this matrix with itself moves all the columns to the right by one column and the last column (which would ‘drop out’) enters the first column. Two other properties are

$$\bar{\mathbf{\Omega}}^T = \bar{\mathbf{\Omega}}^{-1} \tag{5.79}$$

$$\bar{\mathbf{\Omega}}^q = \bar{\mathbf{E}}. \tag{5.80}$$

The eigenvalues of this matrix are

$$\lambda_\Omega(m) = e^{i\frac{2\pi}{q}m} \tag{5.81}$$

where $m = 0 \dots q-1$ and the first eigenvector $|\phi_0\rangle$ is the q dimensional, normalized vector with the elements $1/\sqrt{q}$. Since this eigenvector is also the eigenvector of (5.77) the eigenvalues of $\bar{\mathbf{T}}$ are simply

$$\begin{aligned} t_m &= \omega(0) + \sum_{l=1}^{q/2-1} \omega(l)\left(e^{i\alpha lm} + e^{-i\alpha lm}\right) + \frac{1}{2}(e^{i\alpha\frac{q}{2}m} + e^{-i\alpha\frac{q}{2}m})\omega(q/2) \\ &= \omega(0) + 2\sum_{l=1}^{q/2-1} \omega(l)\cos(\alpha lm) + \cos(\pi m)\omega(q/2). \end{aligned}$$

The correlations (see Equation (5.34)) for the nearest neighbours $l - l'$ can then be found using the operator $\bar{\mathbf{A}}_m$ corresponding to the correlation m

$$\begin{aligned} \langle m\rangle &= \frac{1}{t_0}\langle\phi_0|\bar{\mathbf{A}}_m\bar{\mathbf{T}}|\phi_0\rangle \qquad (5.82) \\ &= \frac{1}{t_0}\langle\phi_0|\frac{\partial}{\partial K_m}\bar{\mathbf{T}}|\phi_0\rangle \\ &= \frac{\omega(0) + 2\sum_{l=1}^{q/2-1}\omega(l)\cos(\alpha lm) + \cos(\pi m)\omega(q/2)}{\omega(0) + 2\sum_{l=1}^{q/2-1}\omega(l) + \omega(q/2)} \\ &= \frac{t_m}{t_0} \end{aligned}$$

Starting from this equation it is possible to analytically find the coupling constants K_m with $m = 1 \dots q/2$, after having measured the correlations $\langle m\rangle$ in the image by multiplying by a sum of cosine terms. In addition it is necessary to fix the value t_0 or $\omega(0)$ since there is one more eigenvalue than there are correlations. This corresponds to fixing the energy of the ground state, and gives

$$\omega(m) = \frac{1 + 2\sum_{j=1}^{q/2-1}\langle j\rangle\cos(\alpha mj) + \langle\frac{q}{2}\rangle\cos(\pi m)}{1 + 2\sum_{j=1}^{q/2-1}\langle j\rangle + \langle\frac{q}{2}\rangle} \qquad (5.83)$$

for the Boltzmann factors and

$$K_m = c\left(2\sum_{j=1}^{q/2-1}\cos(\alpha mj)\ln(\omega(j)) + \cos(\pi m)\ln(\omega(\frac{q}{2}))\right) \qquad (5.84)$$

for the couplings (where $c = 2/q$ if $m < q/2$ and $c = 1/q$ if $m = q/2$).

The energy between two pixels having a brightness (state) difference of Δl is then

$$E(\Delta l) = E(l - l') = \sum_{j=1}^{q/2} K_j \cos(\alpha j(l - l')) \tag{5.85}$$

All of this requires measuring the correlations in the image by running over all the points for the horizontal

$$\langle m \rangle^x_{data} = \frac{1}{N_x N_y} \sum_{x=1}^{N} \sum_{y=1}^{N} \cos\left(\alpha m(l^y_x - l^y_{x+1})\right) \tag{5.86}$$

and vertical bonds

$$\langle m \rangle^y_{data} = \frac{1}{N_x N_y} \sum_{x=1}^{N} \sum_{y=1}^{N} \cos\left(\alpha m(l^y_x - l^{y+1}_x)\right). \tag{5.87}$$

Finally, we obtain the equations for the partition function (which is the largest eigenvalue of $\bar{\mathbf{T}}$)

$$Z = \omega_0 + 2 \sum_{m=1}^{q/2-1} \omega_m + \omega_{q/2}, \tag{5.88}$$

the mean energy

$$\langle E \rangle = -\frac{1}{\beta} \sum_{m=1}^{q/2} J_m \langle m \rangle, \tag{5.89}$$

where $K_m = \beta J_m = \frac{1}{kT} J_m$ and the entropy of the Z_q model

$$H_{phy} = k(\ln(Z) + \beta \langle E \rangle). \tag{5.90}$$

We can set $k = 1/ln2$ for the base-2 logarithm. Note that the β cancels out and we do not require the temperature.

5.4.3 Ashkin-Teller model for grey level images

A common method in predictive image coding is to represent the lattice of 8 bit pixel values as a stack of 8 independent bitplanes, containing only black and white points. The encoding (compression via prediction) step then operates only on separate planes. Translated to the physicists language this means the image can be seen as layers of two-dimensional Ising models as presented in Figure 5.2 (left two columns). The interesting point both from

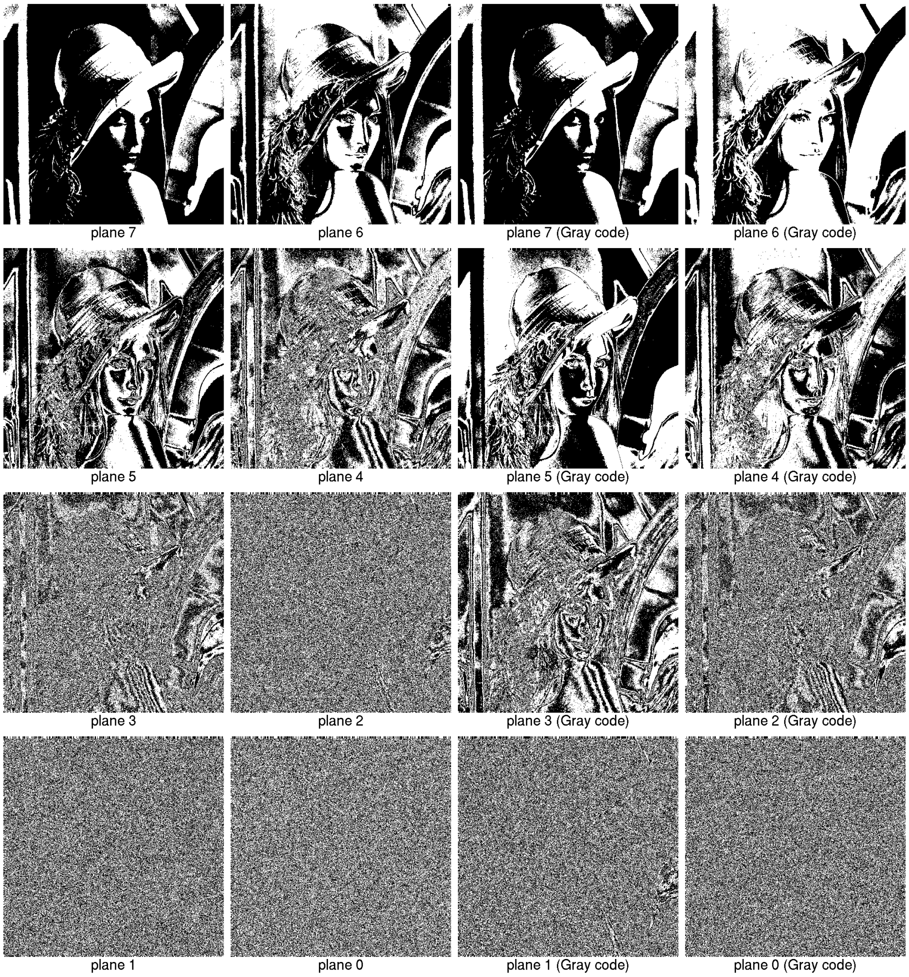


Figure 5.2: Bitplanes of an example image. The two left columns are normal bitplanes, the two right columns are the planes after conversion to Gray coding. Planes 7 correspond to highest bit and planes 0 to the lowest bit.

a physical viewpoint and in terms of predictive coding is that neighbouring spins/pixels in the planes have different innerplane couplings depending on the plane's position. The temperature of a plane increases with decreasing bit position making the spins less and less correlated when one moves to the lower planes. Since the entropy of the four lower order bitplanes is close to 1, these planes cannot be encoded efficiently by compression algorithms and have to be sent almost unchanged. This is why lossless compression typically achieves a maximal rate of around 4 (or 5) bit per pixel (depending on the data). In order to get a better compression rate the image is usually mapped from binary code to Gray code before working with the bitplanes. In the Gray code representation adjacent pixel values only differ by one bit. If – for example – a region in the image fluctuates between values 127 and 128 the binary planes change from 01111111 to 10000000, although the difference in pixel value is small. In contrast, the Gray encoded planes only differ in one bit (from 01000000 to 11000000) As can be seen in Figure 5.2 (right column) the Gray code representation exhibits more 'structure' in the lower planes than the binary coding. The entropy of the planes is lower and the data can be predicted easier resulting in smaller code.

Ashkin and Teller [AT43] introduced a model of a physical system which is actually a Z_q model with $q = 4$ states, with the difference that the whole system is described by two coupled Ising lattices. Couplings exist within each plane (intraplane couplings) and between the planes (interplane couplings). If we do not restrict the number of states, we can encode an image using the AT model. This requires more than 2 bitplanes, typically we will use all 8 planes of the image. The total number of coupling constants is the same as in the Z_q model. An example of the way the planes are coupled is illustrated in Figure 5.3. The index for the couplings K follows the notation in which a set bit in the binary representation of the index value means that this plane takes part in the coupling ($K_2 = K(00000010) \rightarrow$ plane 2, $K_{13} = K(00001101) \rightarrow$ planes 1,3 and 4). In general we use the notation for an index value α :

$$\vec{\alpha} = (b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0) \quad b \in \{0,1\} \tag{5.91}$$

$$\alpha = \sum_{k=0}^{7} b_k 2^k \tag{5.92}$$

$$\vec{\alpha}_k = b_k. \tag{5.93}$$

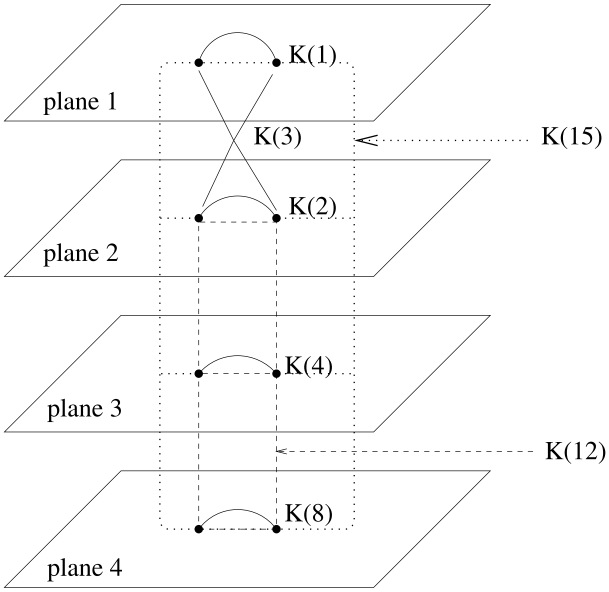


Figure 5.3: Couplings between the different image planes of the AT-model. The example is a model of $q = 16$ states. Some of the couplings and the corresponding planes are shown. The index is calculated from the planes participating in the coupling.

The Hamiltonian for the AT model with p planes ($p = log_2(q)$) is

$$\mathcal{H} = \sum_{\alpha=1}^{2^p-1} K_\alpha \sum_{\langle ij \rangle} \prod_{k=1}^{p} \left(s_i^{(k)} s_j^{(k)} \right)^{\vec{\alpha}_k} \tag{5.94}$$

$s_i^{(k)}$ is the spin state in the plane k at position i. Following the notation of the previous sections, the correlations can be derived by

$$\begin{aligned} \langle \alpha \rangle &= \frac{1}{\lambda_0} \frac{\partial}{\partial K_\alpha} \sum_{k=0}^{2^p-1} \omega_k \\ &= \lambda_\alpha / \lambda_0, \end{aligned} \tag{5.95}$$

where $\{\lambda_j\}$ are the eigenvalues of the transfer-matrix. The couplings can be calculated using

$$\omega_\alpha^{(rec.)} = \frac{1}{2^p}\sum_{\gamma=0}^{2^p-1}\lambda_\gamma\prod_{k=1}^{p}(1-2\vec{\alpha}_k)^{\vec{\gamma}_k} \tag{5.96}$$

$$= \frac{1}{2^p}\lambda_0\left(1+\sum_{\gamma=1}^{2^p-1}\langle\gamma\rangle\prod_{k=1}^{p}(1-2\vec{\alpha}_k)^{\vec{\gamma}_k}\right), \tag{5.97}$$

which results in (choosing $\lambda_0 = 1$)

$$K_\alpha^{(rec.)} = \frac{1}{2^p}\sum_{\gamma=0}^{2^p-1}\ln\left(\omega_\gamma^{(rec.)}\right)\prod_{k=1}^{p}(1-2\vec{\alpha}_k)^{\vec{\gamma}_k}\,. \tag{5.98}$$

This model is more of theoretical interest, since it gives the same results as the Z_q-model for image processing purposes. For this reason no experimental results will be presented for the AT-model in the experiments chapter.

5.4.4 One-dimensional Z_q model for colour images

The ultimate goal in the restoration of still images is the application to full colour images. As explained in the introduction to the data, a colour image consists of three colour planes, each one corresponding to the red, green and blue component of a pixel. The simplest approach would be to treat these three planes independently and apply the Z_q model for grey level images to each one separately. However, this neglects any potential inter-colour interaction between two points. One solution is to use a kind of AT model this time applied not to bitplanes but to colourplanes or to use the AT model with 24 bitplanes. The problem is that the number of coupling constants then would be $2^{24} = 16777216$ which is far more than the image data itself requires for storage. A simpler approach – introduced here – which includes the inter-colour correlations is formulated using the difference between two transformed colour pixels $\tilde{\vec{c}}$ and $\tilde{\vec{c}}'$

$$\Delta_{colour}(\vec{c},\vec{c}') = \sqrt{\sum_{i=1}^{3} w_i(\tilde{c_i}-\tilde{c_i}')^2}, \tag{5.99}$$

where w_i are positive weights for the corresponding components. The transformation itself is usually a linear one, which maps a colour point from the RGB space into another space. One widely used representation is the HSI space where the components are hue, saturation and intensity which separates the colour information from the brightness of the pixel. However, as can be seen in the experiments (Chapter 6), the simplest model in RGB space with all $w_i = 1/3$ already gives excellent results. The remaining problem is to put the difference $\Delta_{colour}(\vec{c}, \vec{c}')$ in a form that can be used in the Z_q model. The main aspect is that the variables in the model belong to the spin-group with finite integer values in the range of $[0, \ldots 255]$, but the colour difference is a real number. The simplest approach is to map this number to the closest integer value which then is a variable of the spin-group. Although this seems a rather crude approximation it is in fact very fast and efficient as can be seen in the results. Hence, the simplest model for two colour pixels that can be used is

$$\Delta^{Zq}_{colour}(\vec{c}, \vec{c}') = int\left(\sqrt{\frac{1}{3}((c_r - c'_r)^2 + (c_g - c'_g)^2 + (c_b - c'_b)^2)}\right), \qquad (5.100)$$

where

$$int(x) = \begin{cases} \lfloor x & \text{if } x - \lfloor x \leq 0.5 \\ \lceil x & \text{else} \end{cases} . \qquad (5.101)$$

This replaces the terms $l - l'$ in the Hamiltonian (Equation (5.50)).

5.5 Restoration of Images

5.5.1 Bayes restoration

After the selection of a model and learning the parameters from the data, the last step that remains is the restoration of the transmitted and distorted image. This procedure, which is based on Bayesian methods coupled with a Markov model, aims at recovering the original data from the received data using the knowledge about the *prior* model (for example the Z_q model from Section 5.4.2) and the model for the channel noise (from Section 3.4). In order to find an estimate of the best restoration, these models are combined into an energy function which is then optimized in the restoration process. Unfortunately, this is a computationally intensive combinatorial optimization problem. However, some methods from statistical physics have proven to be very efficient for this kind of problem.

Metropolis *et al.* [MRR$^+$53] proposed a Monte Carlo simulation to find equilibrium states of a thermodynamical system. It was realized later that there is an analogy between minimizing the energy function of a combinatorial optimization problem and finding the energy minima of thermodynamical systems by cooling the system down until equilibrium is reached [KGV83]. This algorithm was named Simulated Annealing and consists mainly of substituting the energy of a solid by the cost/energy function of the computational problem and then slowly decreasing the temperature. However, one drawback is that SA needs a careful selection of the cooling scheme or otherwise the global optimum will not be found. Several related algorithms were proposed for image processing tasks, the most important one which is known as the Gibbs Sampler was used in the famous paper by Geman and Geman [GG84]. Another solution is to use deterministic algorithms which are not optimal but converge after a few iterations [Li95a, BZYJ96].

The method introduced in this work is based on the simple Metropolis algorithm. The convergence of the restored image is improved in two ways: first by storing at each Monte Carlo step the whole probability of flipping (or not flipping) the spin variable and second by visiting the pixels which have a high local energy with a higher preference. Both will be discussed shortly. This algorithm has the advantage of not being dependend on additional parameters (like temperature), and yet having all the properties of stochasitical optimization methods.

5.5.2 Metropolis-Rosenbluth-Rosenbluth-Teller-Teller algorithm

The method of choice here – the Metropolis algorithm – was already explained in Section 4.4.4 and will be described as a short algorithm again. Let $\vec{r}$ be the received (noisy) image, $\vec{s}^{\tau}$ be the current image undergoing optimization and $\vec{\hat{s}}$ the restored image after convergence. Furthermore, let $E(s_i^{\tau}|r_i)$ be the local energy for the pixel i then the algorithm is :

1. select a pixel i at random
2. calculate $E(s_i^{\tau}|r_i)$
3. choose $s_i'^{\tau} \in [0, \ldots 255]$ at random
4. calculate $E(s_i'^{\tau}|r_i)$
5. let $p = \min(1, e^{E(s_i^{\tau}|r_i) - E(s_i'^{\tau}|r_i)})$
6. replace s_i^{τ} by $s_i'^{\tau}$ with probability p
7. repeat until convergence

Here the probability comes from a Gibbs distribution having the energy as the exponent. These steps are repeated until either a fixed number of iteration steps have been reached or the energy in the image fluctuates around a constant value.

As mentioned, the simple Metropolis algorithm was extended in two ways which are described in the next sections.

5.5.3 Optimal decoding

In the general context of error-correcting codes, it was shown [MMP87, Ruj93] that the average error per pixel is minimized by assigning each decoded pixel the most probable state given by the exact conditional a posteriori probability. This decoding scheme is known in image restoration as 'thresholded posterior mean' [MMP87, PB95] and in the Bayesian literature as 'marginalization'. In the statistical physics setting it corresponds to a finite temperature decoding on the Nishimori line [Ruj93, Sou94]. The error per pixel for the whole image is in this case defined by the mean squared error (Equation (3.33)). In order to minimize the error for the given cost function the optimum $\hat{\vec{s}}$ is found if

$$\hat{s_i} = l \in \mathcal{L} : (\langle s_i \rangle - l)^2 \leq (\langle s_i \rangle - l')^2 \ \forall l' \neq l, \tag{5.102}$$

where $\langle s_i \rangle$ is the average of s_i over all states of the system (ensemble average). In an ergodic system the value of $\langle s_i \rangle$ can be calculated as the average over time (iterations)

$$\langle s_i \rangle \approx \frac{1}{n_t - n_{eq}} \sum_{\tau=n_{eq}}^{n_t} s_i^\tau, \tag{5.103}$$

where n_t is the maximum number of iterations, n_{eq} the number of iterations for reaching the equilibrium and s_i^τ is the state for iteration step τ. Then the state $\langle s_i \rangle$ is the value that will minimize the error per pixel. The new idea used in the experiments is to store the full transition probabilities (of flipping or not flipping the spin variable) at each Monte Carlo step. However, since this requires a large amount of memory this is approximated by calculating the expectation value for each pixel. That is,

$$\langle s_i \rangle \approx \frac{1}{n_t - n_{eq}} \sum_{\tau=n_{eq}}^{n_t} s_i^\tau p(s_i^\tau) / \sum_{\tau=n_{eq}}^{n_t} p(s_i^\tau), \tag{5.104}$$

where p is the probability from the Metropolis scheme and τ the counter for the Monte Carlo steps.

5.5.4 Rank-Order Monte Carlo

The second extension is a 'visiting' scheme for the points in the image which ranks each pixel according to its local energy. Related procedures have been applied to Monte Carlo simulations of the Ising-Model [Wil84]. The pixels with high local energy being potential candidates for flipping are visited with a higher probability. The simplest implementation is to first calculate all local energies (as defined in Equation (5.50)) for the pixels, insert each one into a list of candidates and sort this list. Then the pixels with the highest energy values are selected for the Monte Carlo step. This has the advantage of avoiding those pixels which are unlikely to be flipped and would reject a new state. Especially in the initial phase of the restoration process this speeds up the processing dramatically (see results from Section 6.2.1). However, from a theoretical point of view it is not clear if the system with this scheme is still ergodic. On the other hand, regarding restoration purposes it fulfills two important requirements: during the first iteration phase it is very likely that only those pixels are visited which have a strong distortion (i.e. violate the prior model) and in the phase where the system starts to fluctuates around the equilibrium state the pixels which make up the edges and corners in the image are tested more often. Since these **are** the regions of interest it is well justified to incorporate the rank order scheme here.

5.6 Classification – Texture Segmentation with the Z_q model: a new approach

Texture Segmentation is used to segment an image into different regions according to the textures (intensity structures) of these regions. This is often a necessary preprocessing step to object recognition and detection, where it is assumed that objects in the scene can be distinguished by difference in texture. In supervised texture segmentation the model and parameters are assumed to be known for the textures as well as the parameters for noise. In this case the image can be partitioned according to the textures whose model and distribution functions are completely specified [Li95a, Rus99, BZYJ96]. However, in most cases the parameters of the textures are not known and have to be learned from the data. The problem is that the image has to be partitioned first so that the parameters for the regions can be learned. This can be solved by using an iterative scheme which alternates between estimation and segmentation [Bes86, Li95a].

A new approach introduced here and based on the Z_q model is to calculate for a block of pixels the approximated joint probability of this block and assign this probability to a label image which has at each point a value equal to this probability (for illustration purposes this is multiplied by 255 and quantized). The resulting label image then exhibits regions of different intensities which can be separated by simple histogram thresholding. The model introduced here is based on an independent bonds approximation:

$$P_{block}(x, y) \approx \prod_{k,j \in \mathcal{N}_{x,y}} e^{\mathcal{H}_{Zq}(k,j)}, \tag{5.105}$$

where $P_{block}(x, y)$ is the probability for the block with (x, y) in the center, and the product of the marginal probabilities includes all nearest neighbour pairs. Typical blocksizes are 3×3 or 5×5. Note that the parameters for the Z_q model are estimated in a first step for the whole image, which results in an overall probability sampling. In the second step, the probabilities are assigned to the new labels by setting

$$l_{new}(x, y) = P_{block}(x, y), \tag{5.106}$$

where $P_{block}(x, y)$ from Equation (5.105) contains all possible two-pixel bonds in the block. The Hamiltonian is

$$\mathcal{H}_{Zq}(k, j) = \exp \sum_{m=1}^{q/2} K_m \left(\cos \frac{2\pi}{q} m(l_x^y - l_k^j) \right). \tag{5.107}$$

The histogram of this new label image represents the distribution of these probabilities and any peak corresponds to a feature (class) in the data. In natural images a high label value means that a pixel block belongs to the bulk sample and a low value that it does not. The latter blocks belong mostly to lines and edges. After separating these two (or more classes) the procedure can be iterated, this time learning different sets of parameters for the different regions. Even though the approximation in Equation (5.105) is rather simple it gives similar results to methods like Sobel or variance filtering [Rus99]. A more advanced procedure which aims at capturing the true underlying probability density is introduced in Chapter 7.

5.7 Noise estimation and model verification

An open challenge for the Bayesian approach to image restoration is to find the full a posteriori probability (Equation (4.17)) which also includes a model of and parameters for, the noise. As long as the noise process and the strength of the noise is known, the statistical model of the channel can be found in most cases. In some cases it is possible to find the model from the known underlying physics of the transmission, but the estimation of the parameters may still not be easy. The usual method for determining the characteristics of a system is to send a set of exactly known phantom (test) images through the channel and study the statistics of the received set. For example, if after a transmission a plain white image shows a Gaussian intensity distribution then the channel noise is very likely to be additive white noise.

Another idea is to use the known prior of the data to estimate the noise. Assume that the prior model and the parameters are known (measured before transmission or measured in a set of undistorted images), then for each point the true pixel value l_i should be **exactly** the same as a value $\hat{s_i}$ predicted by the model. For the Bayesian approach this means that

$$l_i \overset{!}{=} \hat{l}_i = \text{arg max}_{l_i \in L_i} P(l_i | l_1, \ldots, l_{i-1}, l_{i+1}, \ldots, l_N), \tag{5.108}$$

where $P(l_i|\vec{l}, l_i \notin \vec{l})$ is the prior of the image leaving out the pixel l_i. In practice this prior is restricted to the points in the (Markov-) neighbourhood of l_i and this becomes $P(l_i|l_j, j \in \mathcal{N}_i)$

To investigate the properties of this estimate it is instructive to use a three dimensional histogram where each bin is addressed by the true value and the estimated value. If there is no noise the histogram has entries only on the diagonal (both values are equal) and the shape of the histogram along this diagonal is the unaltered brightness histogram of the image. This is shown in Figure 5.4, which contains the histogram of the portrait (from the data section, Figure 3.1). Any noise will now enter as off-diagonal elements in this plot. If the shape of the projection perpendicular to the diagonal is not constant along the diagonal then the noise will additionally be brightness dependent (which is, for example, the case in X-ray film). For the case of additive white noise the parameter σ can be determined by measuring the standard deviation of the projection.

We can also apply this procedure to the original image itself and therefore verify the model and its parameters. If model and original image fit perfectly then the estimation should be exactly the same as the true pixel value. Any deviation is either a sign of an imperfect model (the pessimistic view) or shows that the original itself was already subject to noise (the optimistic view). It is interesting to see that the standard test images used in the experiments appear to belong to the latter view, since an attempt to restore the original one – assuming it is noisy – actually **increases** the (subjective) quality of the image. Note that we can't measure this, since again we don't have the true original for comparison.

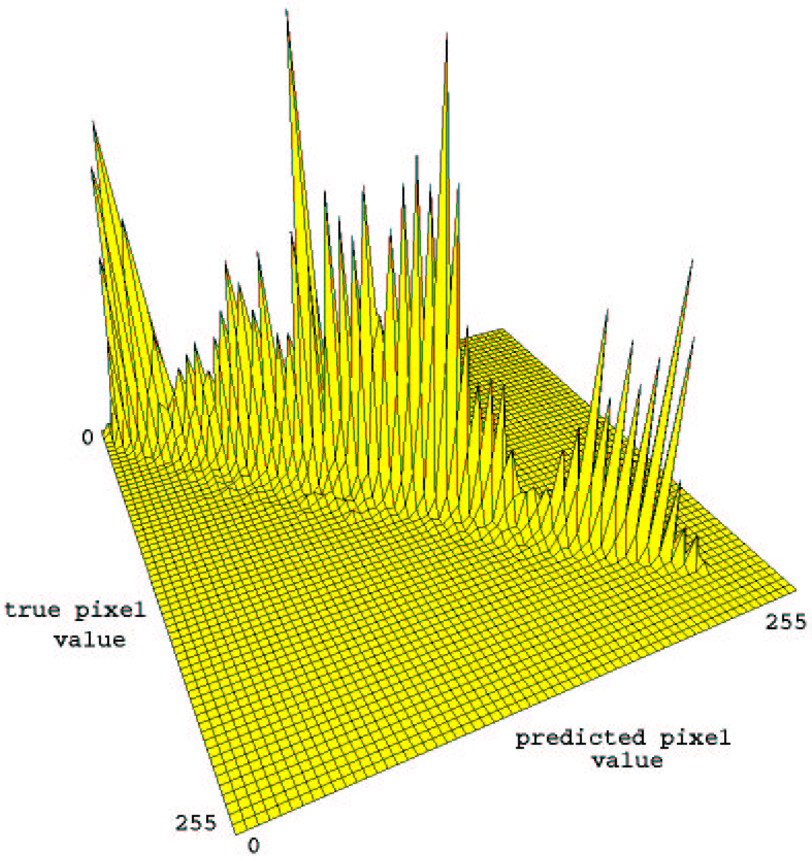


Figure 5.4: Three dimensional histogram from image in Figure 3.1 (portrait). Each bin is addressed by the true pixel value and the predicted pixel value from the neighbourhood.

Chapter 6

Experiments and Applications

This section presents a variety of experiments on a set of test images to illustrate the properties and characteristics of the model. The noisy transmission is simulated with the noise models described in Section 3.4.

6.1 Two dimensional Ising model

We first consider the simple example of a 2D-Ising model in which the parameters for the black and white image displayed in Figure 6.1a are learned using the 2D variational Ansatz. Since the entropy of the scene yields a value of 0.064 bit per pixel, a corrupted image with a channel noise level of $p = 0.35$ (probability of spin flip) can still be restored. The distorted and the restored (Bayes) image are shown in Figure 6.1b and Figure 6.1d. After 180 Monte Carlo steps the error per pixel has decreased from 0.35 to 0.03 (Figure 6.2).

However, the curved object in the lower right corner vanishes, indicating that the assumption of a global prior does not hold here. To test this and to see whether the Monte Carlo dynamics move the image away from the original, we ran the simulation *without* the coupling to the data using only the prior. As can be seen in the graph in Figure 6.3, the error increases with the number of steps. The resulting Bayes image and a Monte Carlo snapshot after 1000 Monte Carlo steps are shown in Figure 6.4.

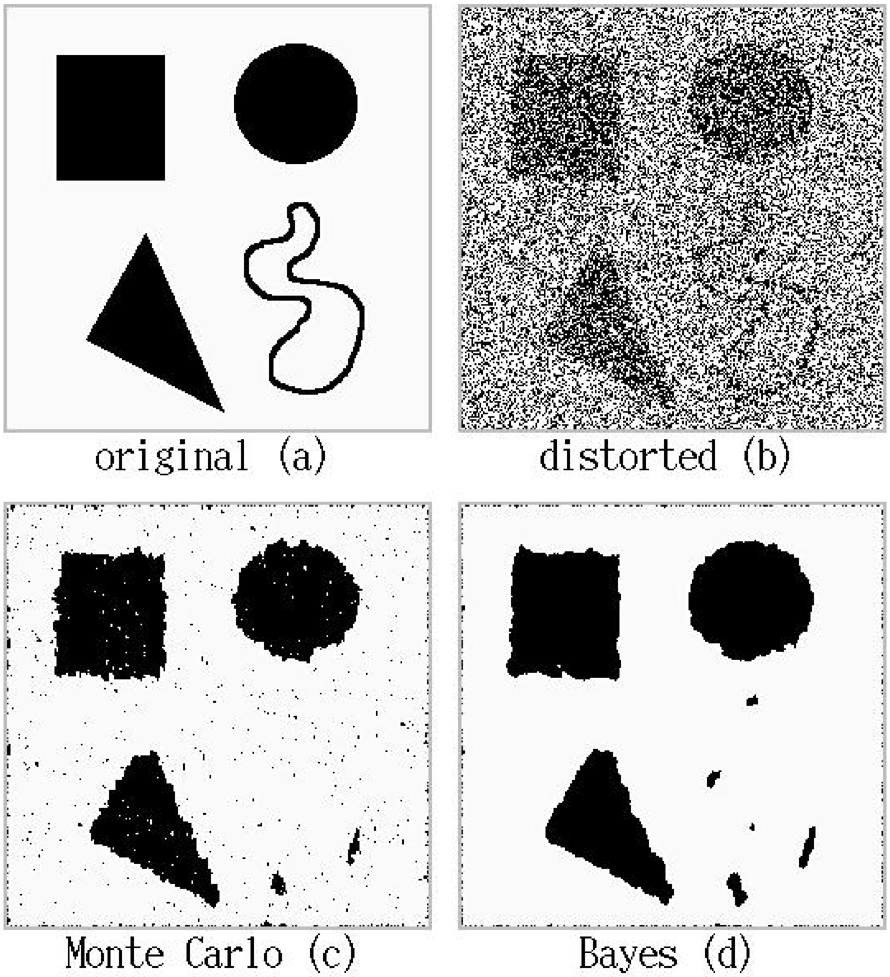


Figure 6.1: Black and white image described by a two-dimensional Ising model. The original (a) is distorted by random noise (single-spin-flip) with a probability of $p = 0.35$ (error). Image (c) is a snapshot of the Monte Carlo image (error 0.04) after 180 steps and (d) the resulting Bayes image (error 0.03).

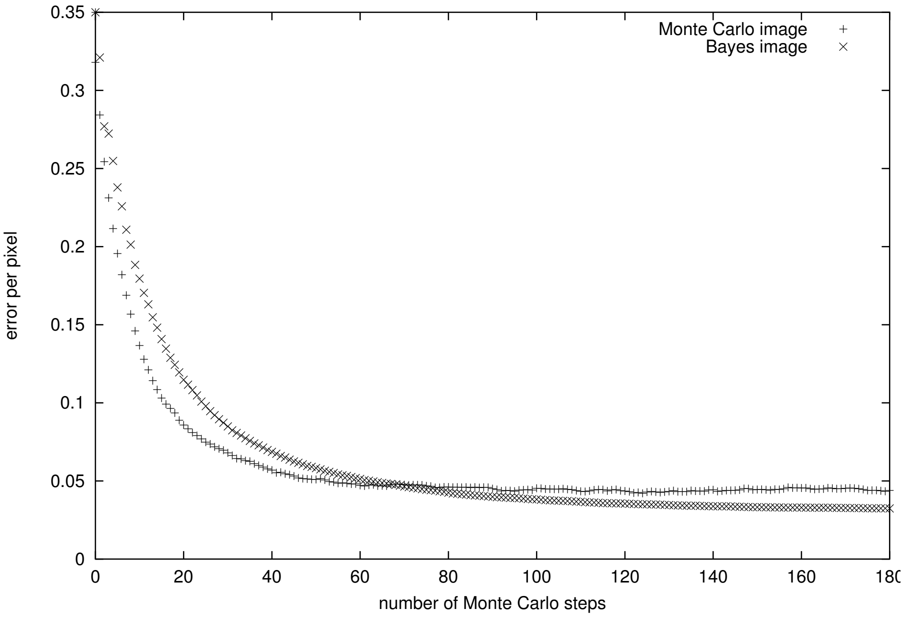


Figure 6.2: Error vs. Monte Carlo steps of the Monte Carlo (+) and Bayes (×) image for the restoration in Figure 6.1. (The error is defined as the fraction of pixels differing from the original.)

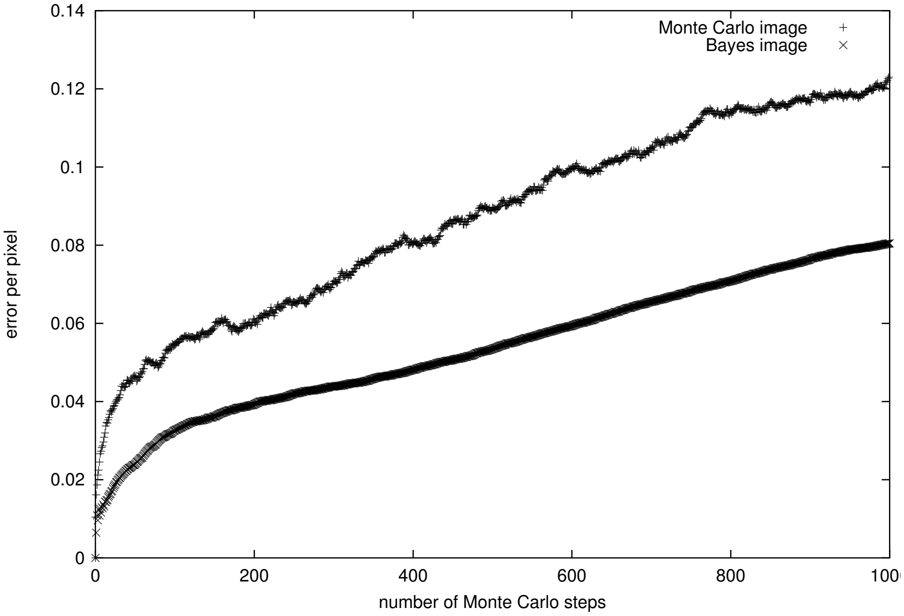


Figure 6.3: Monte Carlo dynamics of the original Ising image (Figure 6.1a) without any coupling to the data (prior only). Displayed is the error vs. Monte Carlo steps of the Monte Carlo (+) and Bayes (×) image.

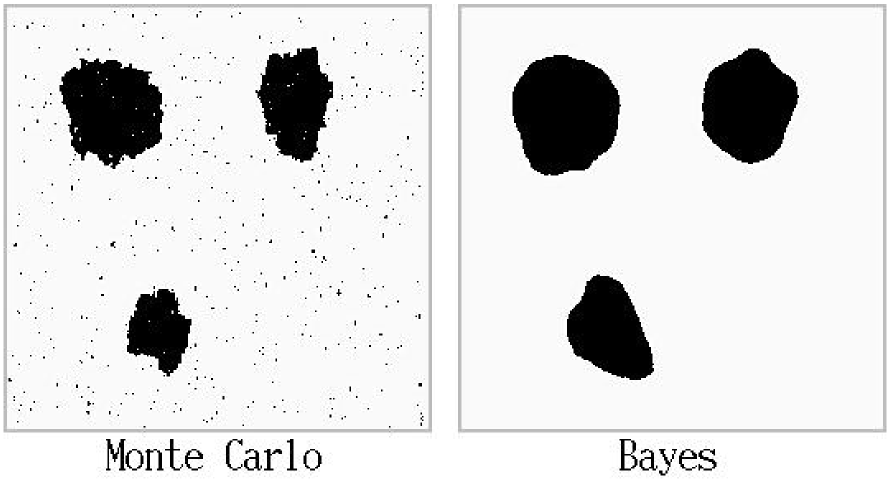


Figure 6.4: Monte Carlo snapshot (error 0.12) and Bayes image (error 0.08) of the prior-only dynamics (no coupling to the data) after 1000 Monte Carlo steps.

6.2 One dimensional Z_q model

The following sections provide an overview of the experiments based on the Z_q model.

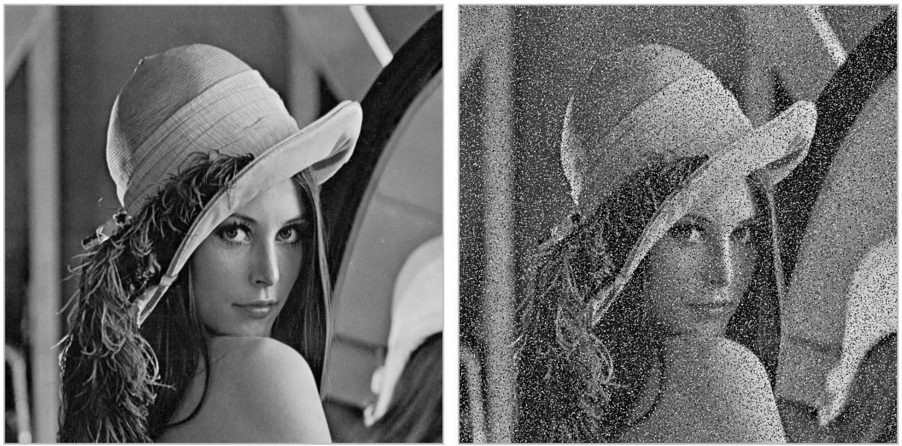
6.2.1 Correlations and random noise

Next we consider natural grey-level images with 256 states per pixel (8 bit image). As an example the famous Lena image in Figure 6.5a is sent through a channel which turns each pixel value independently into any of the other 255 states with a probability of $p = 0.3$ (see Figure 6.5b). The values of the observables derived in the Z_q model are displayed in Figure 6.6 – showing the values for correlations, Boltzmann factors, coupling constants and energy respectively. Using the knowledge about the noise process we obtain the Bayes estimate of the optimal reconstruction in Figure 6.5d. Figure 6.5c contains a Monte Carlo snapshot after 750 steps. The rms error per pixel is reduced from its initial value of 48.51 to 9.01. However the minimal error of 8.36 is reached after 180 steps and then goes up again (see Figure 6.7).

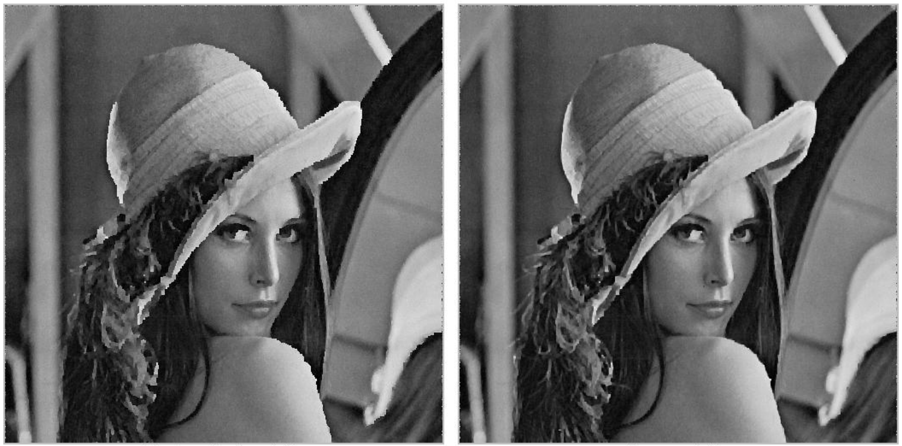
6.2.2 Rank-Order scheme

As a measure of convergence, we count the number of actual pixel 'flips' per pixel during one Monte Carlo step. The graph in Figure 6.8 indicates that the main part of the restoration process takes place in the first 100 steps. After the 150th step the fraction of changed pixels fluctuates constantly around a value of 0.014, but does not contribute much to the restoration. In fact, the error in the Bayes estimate increases after 180 Monte Carlo steps, whereas the error in the Monte Carlo image estimation already goes up again around Monte Carlo step 70.

To explicitly select the pixels which have a high probability of being flipped the rank-order scheme from Section 5.5.4 is used in the same experiment. Figure 6.9 compares the simple Monte Carlo procedure which visits each pixel to the same probability with the rank-order method which prefers pixels that have a higher energy. The fraction of pixels used here was 0.2 of the whole population. This means the algorithm only visits the 20 percent of sites which have high energy. It can be seen that the minimum error is reached after 30 iterations where the simple method takes more than 120 steps. However, in the rank-order scheme the error starts to increase after 40 steps. The reason is that after the noise has been removed, the pixels



original (a) distorted (b)



Monte Carlo (c) Bayes (d)

Figure 6.5: Grey-level image (256 states, 8 bit) distorted by random noise with a probability of $p = 0.3$. The displayed images are: (a) original, (b) distorted (rms=48.51), (c) Monte Carlo snapshot (rms=11.83) and (d) Bayes estimate (rms=9.01) after 750 steps.

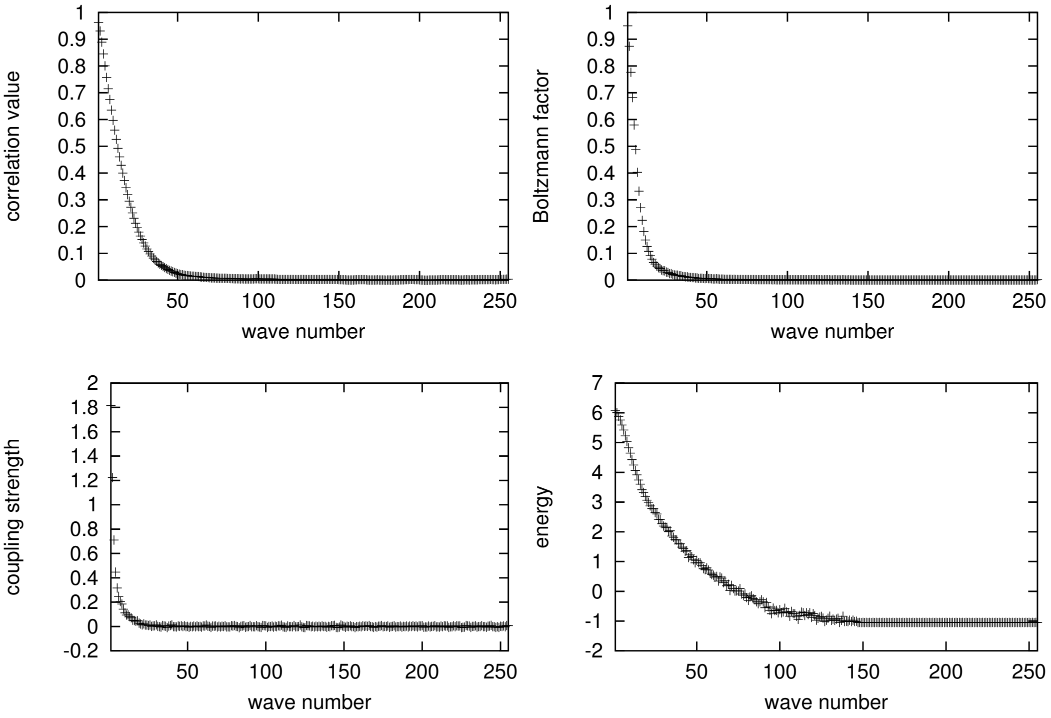


Figure 6.6: Observables – based on the Z_q model – measured in the grey-level image in Figure 6.5a. Displayed are: nn-correlations (defined in Equation (5.82), upper left), Boltzmann factors (Equation (5.83), upper right), coupling constants (Equation (5.84), lower left) and coupling energy (Equation (5.85), lower right). Note that the energy enters the exponent of the Boltzmann distribution with a negative sign.

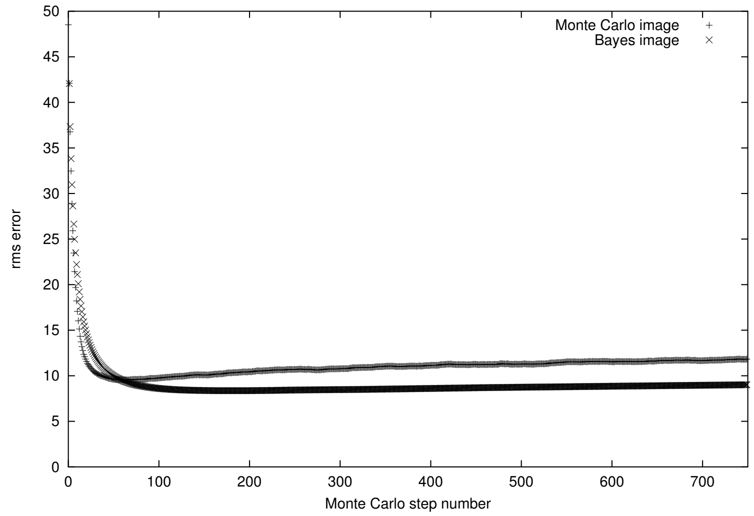


Figure 6.7: Error vs. Monte Carlo steps of the Monte Carlo (+) and Bayes (×) image for the restoration of the grey-level picture in Figure 6.5. Initial rms error is 48.51 ($p = 0.3$). (The error is defined as the root mean square (rms) pixel difference error from the original.)

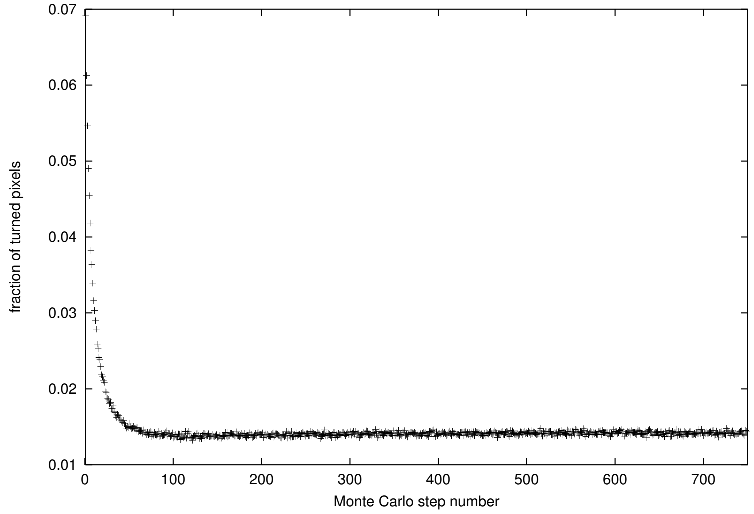


Figure 6.8: Fraction of turned pixels (accepted in Metropolis decision) during a Monte Carlo step over number of steps. Restoration was run for the grey-level image Figure 6.5.

selected to be flipped are those belonging to the lines and edges which have higher energies. These are not noisy pixels, but when changed they start to distort those regions.

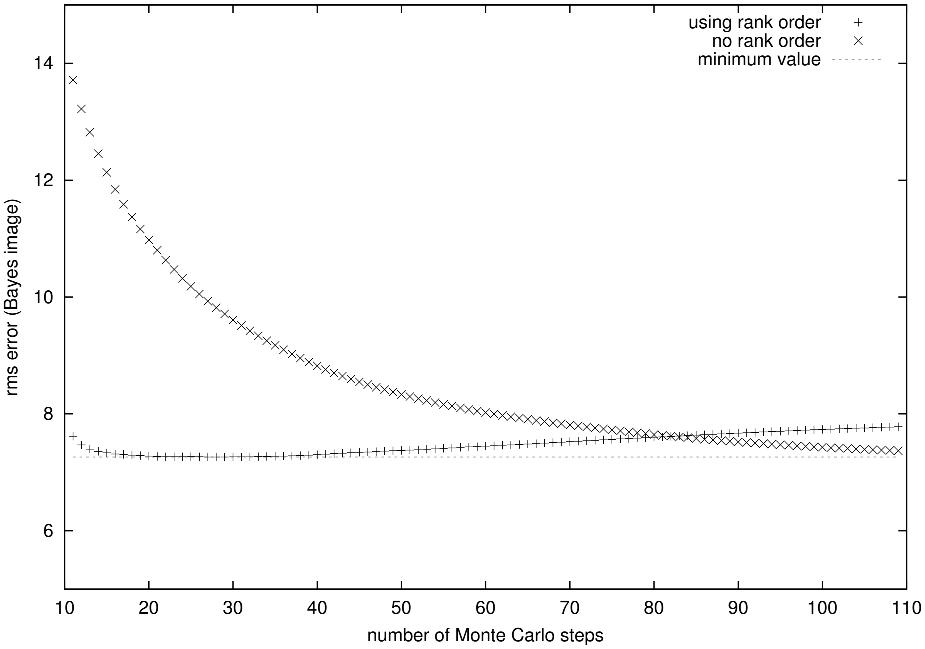


Figure 6.9: Comparison between restoration with the rank-order scheme and simple Monte Carlo. The dashed line is the minimum error reached after 30 Monte Carlo steps.

6.2.3 Close to the prior information

To see how the system performs for an information loss close to the prior information of $H = 4.4$, the image is sent through a channel with a noise probability of $p = 0.45$ (Figure 6.10a). Figure 6.10b shows the restored image after 750 Monte Carlo steps. Again, the rms error fails to decrease after 180 steps (Figure 6.11). Yet, the visual quality still increases which is mainly caused by the breaking of small 'ferromagnetic' islands – clusters of connected pixels which were not turned during the first steps. These clusters disturb the visual quality more than the increasing overall error can take into account (see Figure 6.11). The rms error goes down from 59.27 to 10.6.

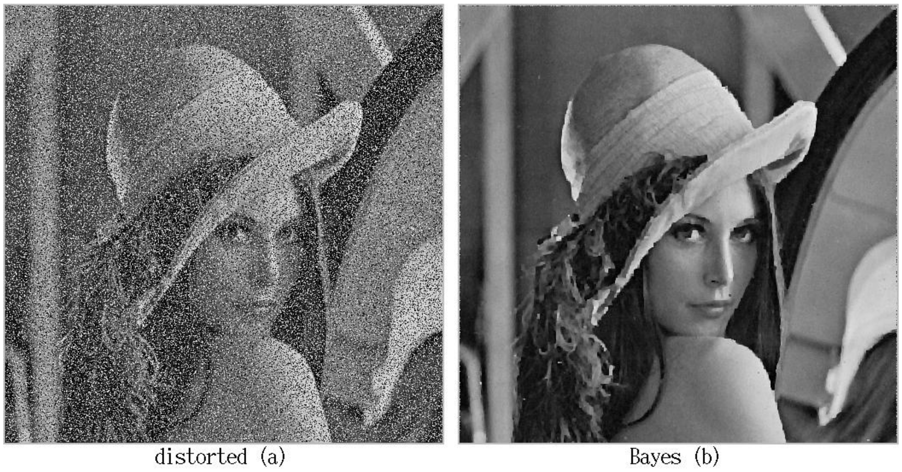


Figure 6.10: Image distorted by random noise with $p = 0.45$ (rms=59.27) (a). The Bayes restoration (rms=10.6) (b) is obtained after 750 steps. The information loss due to noise is close to the information contained in the prior.

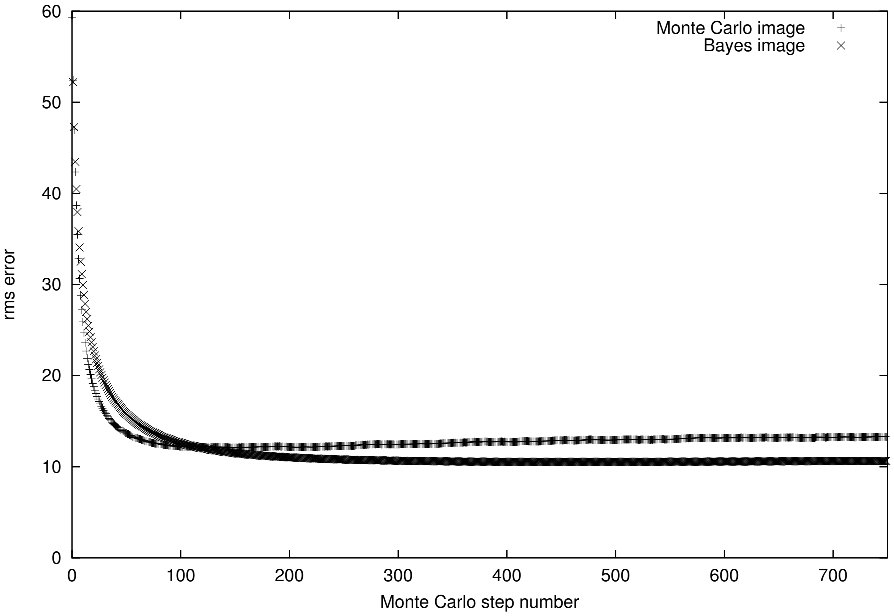


Figure 6.11: Error vs. Monte Carlo steps of the Monte Carlo (+) and Bayes (×) image for the restoration of the grey-level picture in Figure 6.10. Initial rms error is 59.27 ($p = 0.45$).

6.2.4 Prior-only Monte Carlo

Again we are interested in the no-noise dynamics of the system, that is without coupling to any data. The simulation results over 100000 Monte Carlo steps are displayed in Figure 6.12. It becomes clear from the two snapshots of the Monte Carlo and Bayes image after 25000 and 100000 steps (Figure 6.13) that the image moves away from the original and 'condenses' into the state of lower energy (or higher probability), where long range correlations persist; thus we expect that the system is in a temperature regime below its critical temperature.

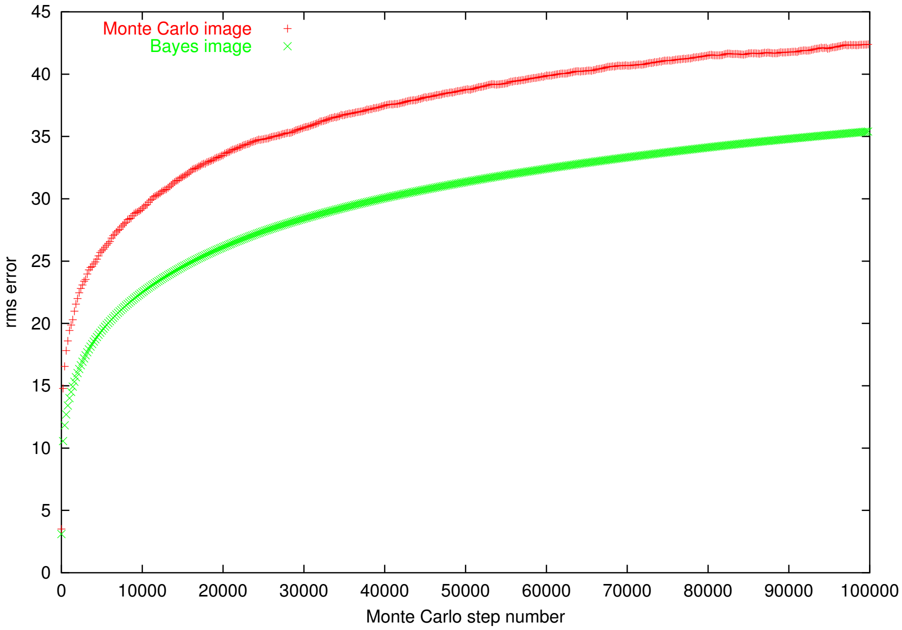
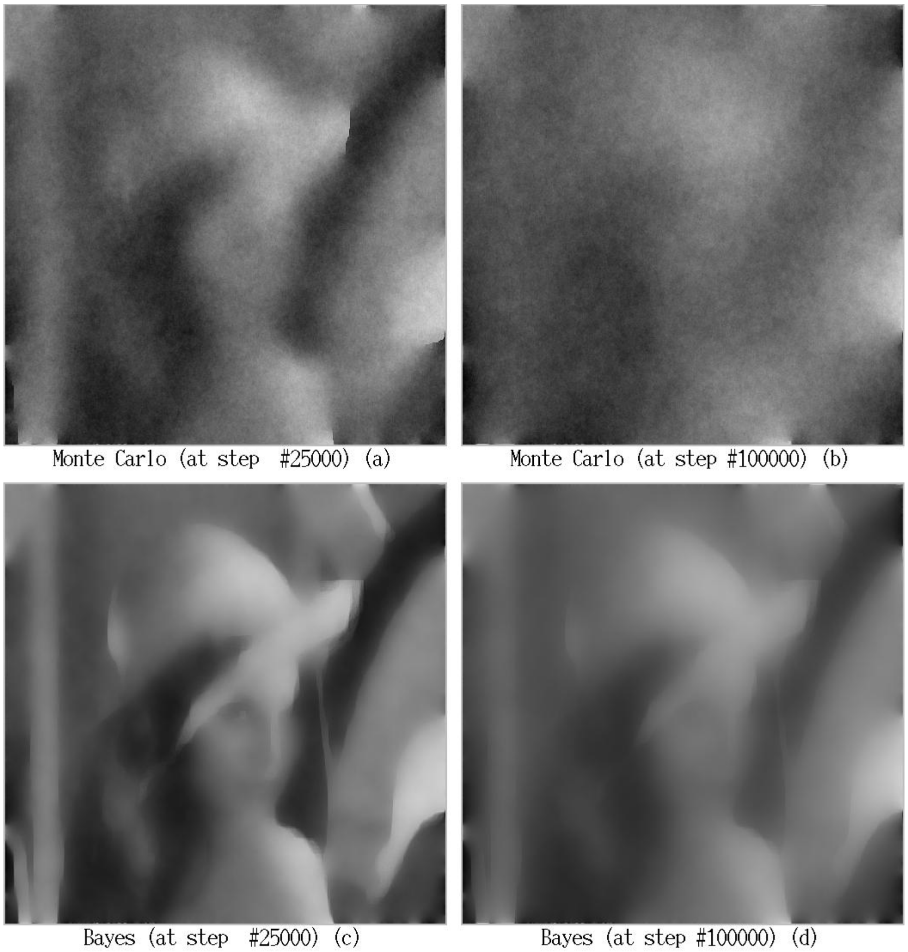


Figure 6.12: Error vs. Monte Carlo steps of the Monte Carlo (+) and Bayes (×) image for the Monte Carlo dynamics of the grey-level picture from Figure 6.5a without coupling to the data (prior only).

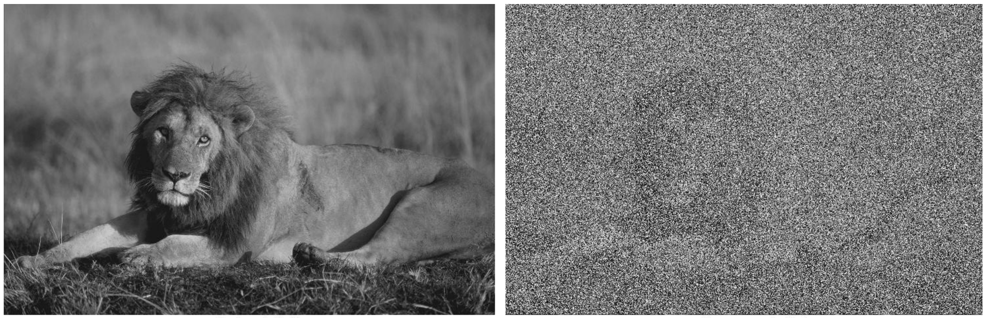


Monte Carlo (at step #25000) (a)
Monte Carlo (at step #100000) (b)
Bayes (at step #25000) (c)
Bayes (at step #100000) (d)

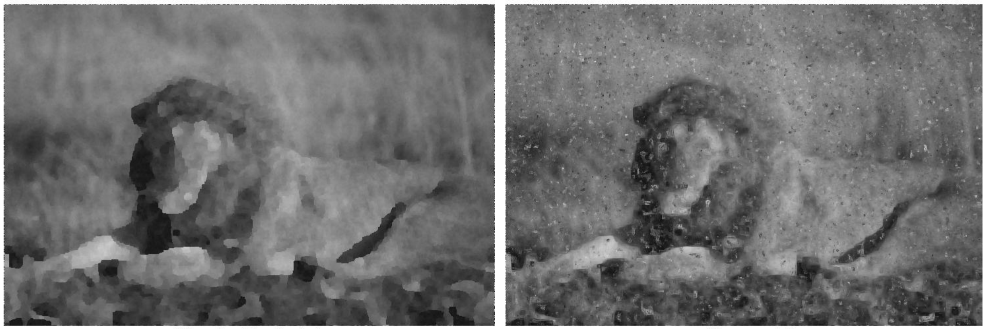
Figure 6.13: Snapshots of the Monte Carlo and Bayes image after 25000 (a and c) and 100000 (b and d) Monte Carlo steps for the simulation without coupling to the data (prior only).

6.2.5 Random noise – high noise regime

The performance of the restoration for the high noise regime is examined in the example shown in Figure 6.14. The displayed image is distorted by random noise of $p = 0.8$ and restored in 1500 iterations. Even though the contents of the image are hardly recognisable in the noise image, the main features are recovered in the Bayes restoration.



original (a) distorted (b)



Monte Carlo (c) Bayes (d)

Figure 6.14: Image corrupted by random noise $p = 0.8$. The restoration was obtained after 1500 Monte Carlo steps (without rank-order scheme).

6.2.6 Learning from the noisy observation

In real life image restoration the original image is usually not available and the parameters have to be determined from the noisy observation (data). Although optimization techniques exist which combine restoration and estimation of parameters [Li95a], we try the very simple way of learning the parameters directly from the noisy image. This shows how sensitive the procedure is to changes of the coupling constants. The bridge image in Figure 6.16a was distorted by random noise with a probability ranging from

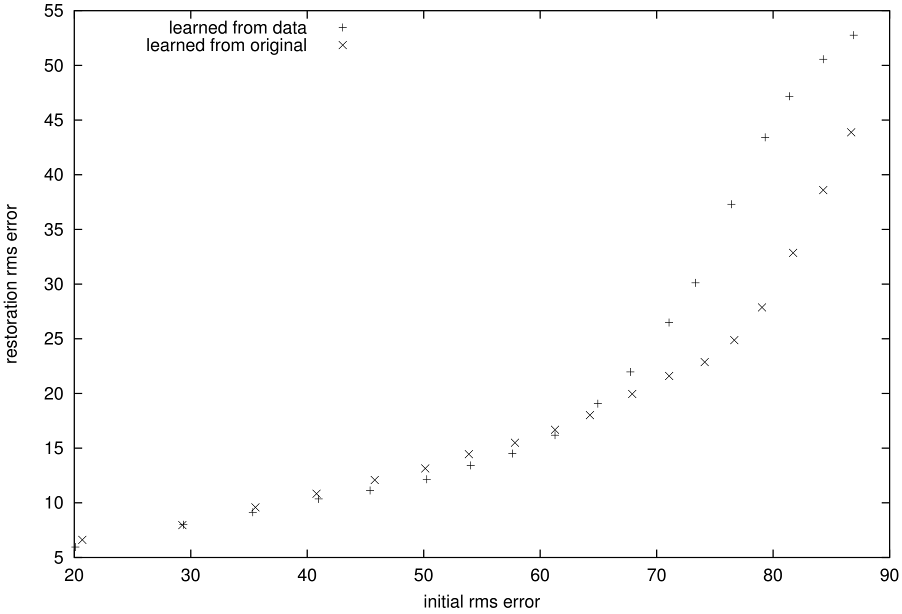
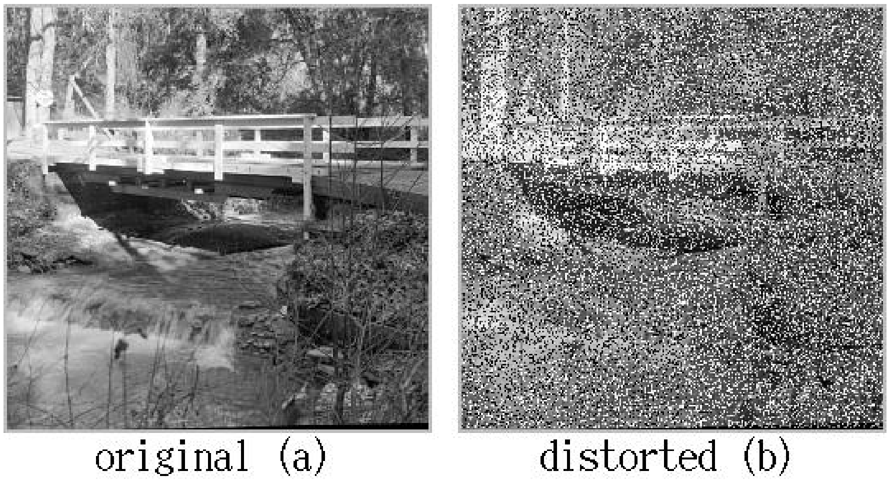


Figure 6.15: Dependence of restoration rms error (in Bayes image) on the initial rms error. The parameters for the prior are learned from the noisy data (+) and – for comparison – from the original (×).

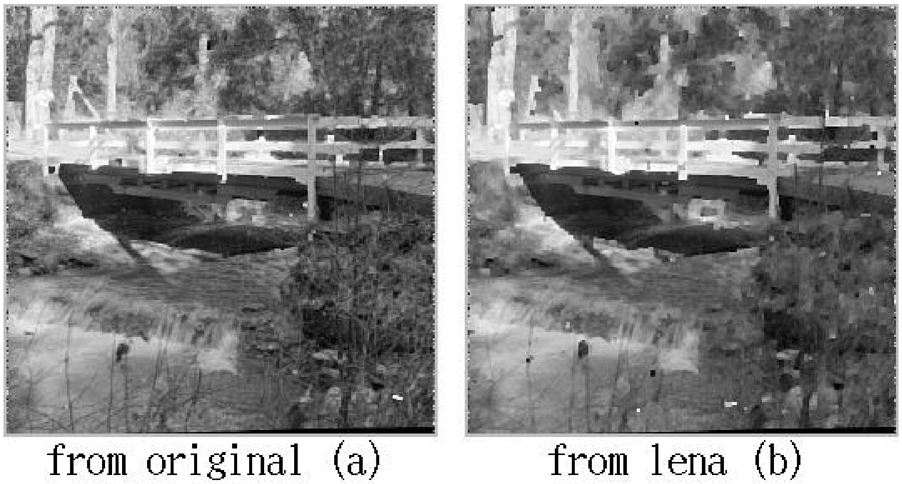
$p = 0.05$ to $p = 0.95$. The couplings were learned from the data and – for comparison – from the undisturbed source. The rms error of the restoration is plotted in Figure 6.15. As can be expected the error first increases linearly with increasing initial error and goes up around the value of $p = 0.625$ which corresponds to the entropy of the original image $H(prior) = 5.951$ (calculated from the prior using Equation (5.90)).

The difference in the restoration between the two sets of couplings is not very large and we are led to ask the question whether the exact values of the couplings are crucial, or if the model we have chosen is already general enough to capture the important statistics of natural images as long as the couplings are kept within a certain range. To test this, we learn the parameters from the Lena (Figure 6.5a) image and use them for the restoration of the bridge image in Figure 6.16, this time distorted by an impulse noise which adds or subtracts a value of 100 to/from each pixel with a probability of $p = 0.3$. The result of the restoration with parameters from the original and Lena are displayed in Figure 6.17. The rms error of the distorted image is 66.47 and goes down to 14.10 for the original parameters and 17.81 for the



original (a) distorted (b)

Figure 6.16: Image of a bridge distorted by impulse noise (b) with probability of $p = 0.3$ (rms=66.47). The height of the impulse is ±100.



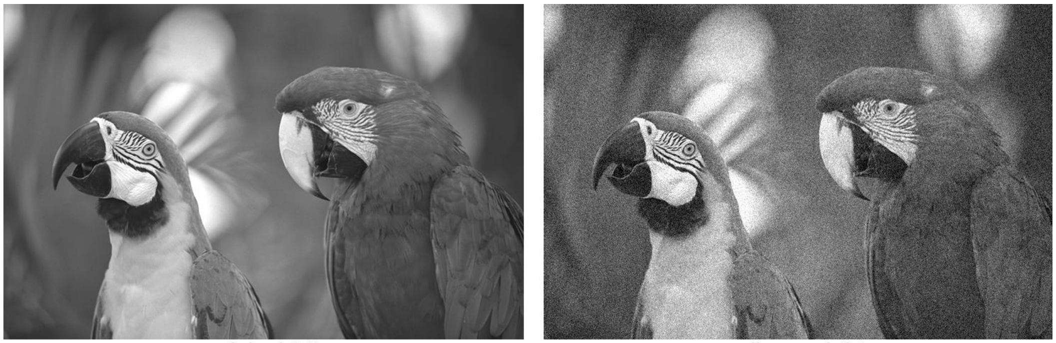
from original (a) from lena (b)

Figure 6.17: Reconstructed image from Figure 6.16b. The prior parameters for the restoration in (a) are learned from the original (rms=14.10) and for (b) from the lena image (Figure 6.5a) (rms=17.81).

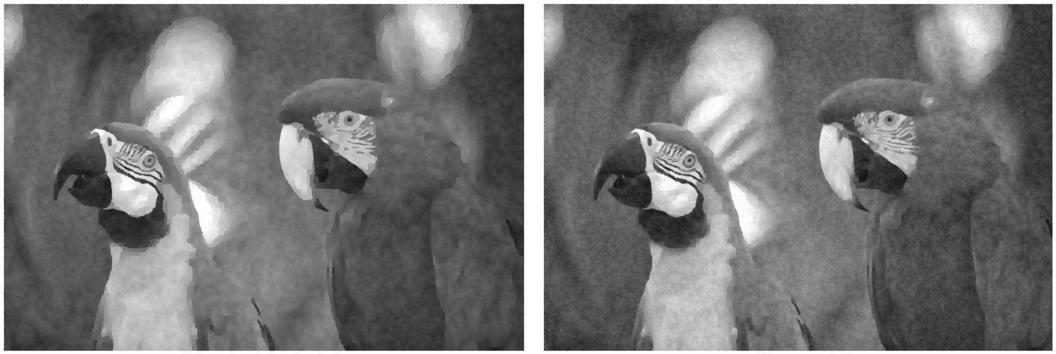
set from Lena. Although there is in both cases a significant improvement, the restoration with the 'Lena'-set of parameters is smoother than with the original set. The reason is that the couplings in the Lena image are more 'ferromagnetic' – they belong to a lower temperature.

6.2.7 White noise

A type of noise which can be found very often in natural or electronic systems is Gaussian white noise. In the next experiment the parrot image (Figure 6.18a) is degraded by white noise which adds a value with zero mean and $\sigma = 20$ to each pixel resulting in a total rms error of 19.837. After restoration this error decreases to 8.716 and the white noise is removed (Figure 6.18d). However, the restoration is much smoother than the original and in comparison to the noisy image the optical 'impression' is not significantly enhanced. The reason for this is that white noise does not 'break' the next-neighbour correlations like random noise and it is this kind of correlation which is important to the human visual system. For comparison, another 'state-of-the-art' restoration method was applied, which is based on a diffusion model (see [EFF93] for details). This algorithm achieves an rms error of 9.126.



original (a) distorted (b)



Monte Carlo (c) Bayes (d)

Figure 6.18: Restoration of data distorted by additive white noise with $\sigma = 20$. Images are: original (upper left), noise image (upper right, rms=19.837), Monte Carlo snapshot (lower left, rms=8.776) and Bayes image (lower right, rms=8.716).

6.2.8 Masking and the human visual system (HVS)

This leads to the interesting question whether the human visual system (HVS) has implemented the knowledge about the correlations which are present in all natural images. If this is the case, then it is possible to degrade an image in a very simple way by inverting the next neighbourhood correlation resulting in a perceptually very 'bad' image. Figure 6.19 (top row) shows an example of an artificially constructed image with 256 grey-levels. The coupling between two points were chosen from a natural image and then simply inverted by setting the new $K'_n = K_{255-n}$. The image is a result of a Monte Carlo simulation after 500 steps, where the starting point was a random image. From a physical point of view the image 6.19 (top right) is not random, since it has clearly defined correlations, however no structures can be **seen**. In order to break up the "bad" correlations we reduce the size of the image and assign to each point of the new image the mean value of a block of four pixels – this is similar to a renormalization process. Now the structures become visible, since the strong anti-correlations are smoothed out.

The same effect can be achieved with a true image by simply replacing every second pixel with its value subtracted from the maximum value (Figure 6.19, lower left). Since every pixel value can be recovered exactly, no information is lost in the masking procedure. Note that the printed version is not optimal to illustrate this effect. The printer can only use black and white pixels and has to combine several of these for a grey-level value. This is in most cases achieved by a process called 'dithering' which distributes the pixels in a very small area randomly, hence the anti-correlations are broken up again.[1]

---

[1] We can see that our vision is tuned to work with these correlations and breaking them might result in a much slower perception. Talking about getting funny ideas when writing down a thesis. The following example shows how we are also tuned to extract patterns from written text (here the neighbour word/character correlations are altered): **Thiss entencei sa lmosti denticalt ot heo riginalb uti tt akesm uchm oret imet or eadi ta ndi ti sq uiteh ardt od o af asts cant og raspt hem eaning.**

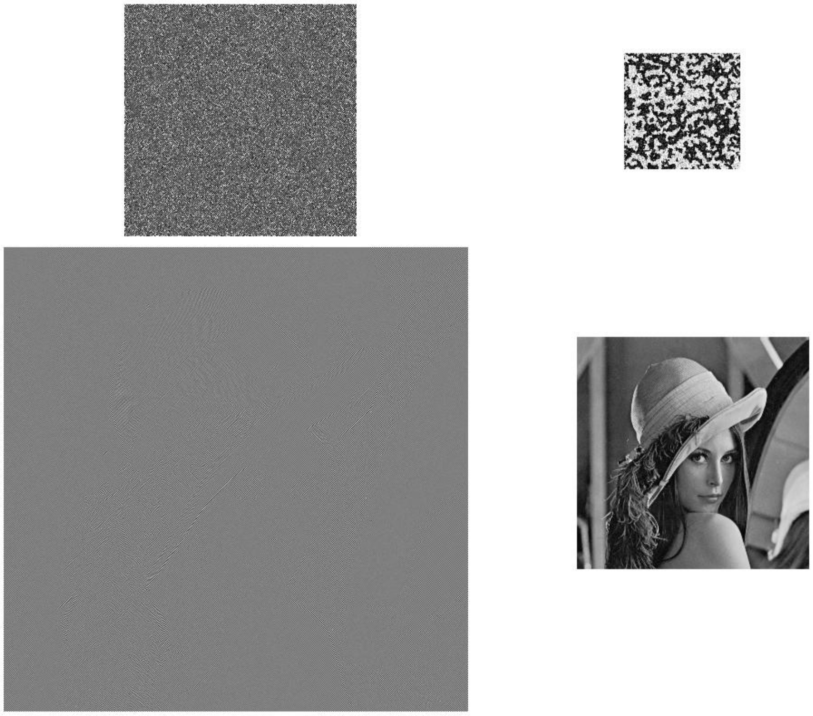


Figure 6.19: Set of images demonstrating that next neighbour correlations are important to the HVS. Top row : artificially created image (Z_q model from a Monte Carlo simulation) (left), the image reduced by a factor of four (right). Lower row : natural image ‘masked’ by replacing each second pixel with its inverse (left) and the reduced image (right).

6.2.9 Deblurring

So far all experiments had to deal with pixelwise and hence spatially uncorrelated noise. However, the model can also restore data which was subject to correlated noise. The left image in Figure 6.20 was blurred by applying a 7x7 pill-box blur (with all components set to 1/49). As was explained in Section 3.4.2 this filter is not exactly invertible by a simple deconvolution due to the quantization. The quantization error measured in bits per pixel for this image is 5.614 bpp. The initial error for the blurred image is 12.295 and goes down to 9.195. From a subjective point of view the image also looks 'sharper' than the distorted image, although the restoration process is unable to restore fine details present in the original image. These details have completely vanished in the distorted data and cannot be recovered.

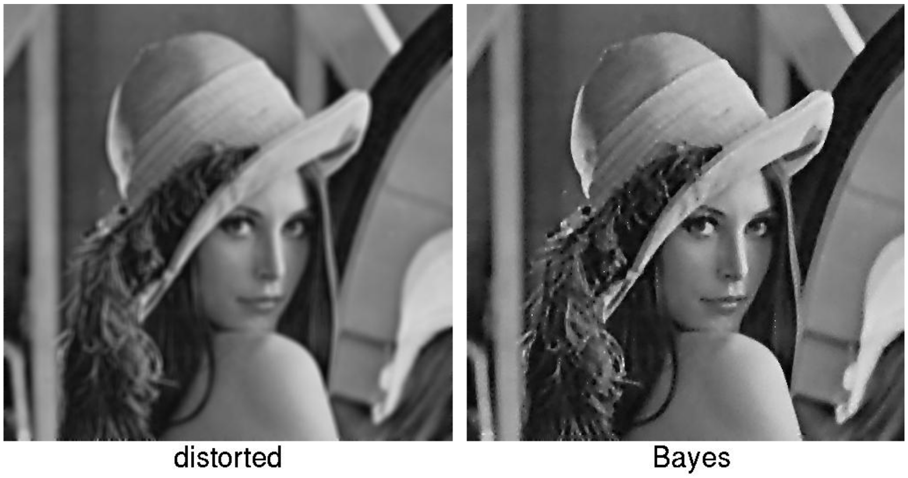


Figure 6.20: Example of debluring. The left image shows the effect of applying a pill-box blur filter of size 7x7 pixels. The right image shows the restoration.

6.2.10 Colour images

Finally, we use the model for colour images in which a pixel is stored as a red, green and blue triplet, each component having 256 states. In this case we use the model introduced in Section 5.4.4 and define the energy as

$$E(\vec{c}, \vec{c}') = \sum_{m=1}^{255} K_m \cos\left(\frac{2\pi}{510} m |\vec{c} - \vec{c}'|_2\right), \tag{6.1}$$

where $|c(\vec{i}) - c(\vec{i}')|$ is the normalized and rescaled euclidian distance of two neighbouring pixels in RGB space. Note that this value is quantized back to 256 discrete levels. The result of random noise with a probability of $p = 0.6$ per pixel and channel (with a total number of wrong colour pixels of $p = 0.92$) is displayed in Figure 6.21. The initial rms error of 129.1 (measured in RGB space) drops down to 19.2 for the Bayes estimation.

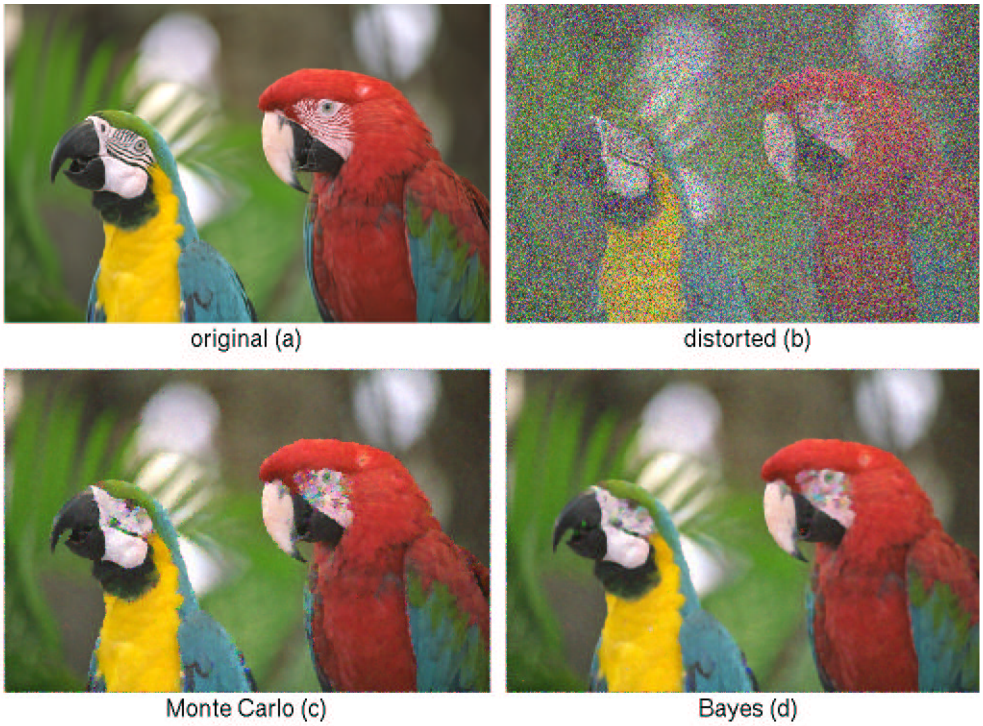


Figure 6.21: Colour image (256 states, 8 bit per colour component) distorted by random noise with a probability of $p = 0.6$. The displayed images are: original (upper left), distorted (rms=129.1, upper right), Monte Carlo snapshot (rms=23.3, lower left), Bayes estimate after 900 steps (rms=19.2, lower right).

6.3 Noise estimation

The information about the correct noise model and strength is not always available. In some applications it has to be estimated from the data. The procedure explained in Section 5.7 is applied to the Lena image in order to verify the model itself. The first plot in Figure 6.22 (+) shows the distribution of differences between the predicted value (from the model) and the true pixel value. The four surrounding neighbour pixels are used to predict the middle point. This model verification has an error (assuming a Gaussian distribution) of $\sigma_{model} = 4.648$. Hence, the model can be used as a predictor for compression, but does not capture all the information. This would require a model with a larger neighbourhood. To see how the model performs for noise estimation, the same procedure is applied to the image distorted by additive white noise with $\sigma_{noise} = 12.0$. The expected value for the noise image should then be the sum of the two sigmas $\sigma_{expected} = \sqrt{\sigma^2_{model} + \sigma^2_{noise}} = 12.869$ (shown in the third plot with dashed line). However, the deviation of the measured noise distribution (shown in the Figure with (×) is $\sigma_{measured} = 14.498$, which is higher than the expected value.

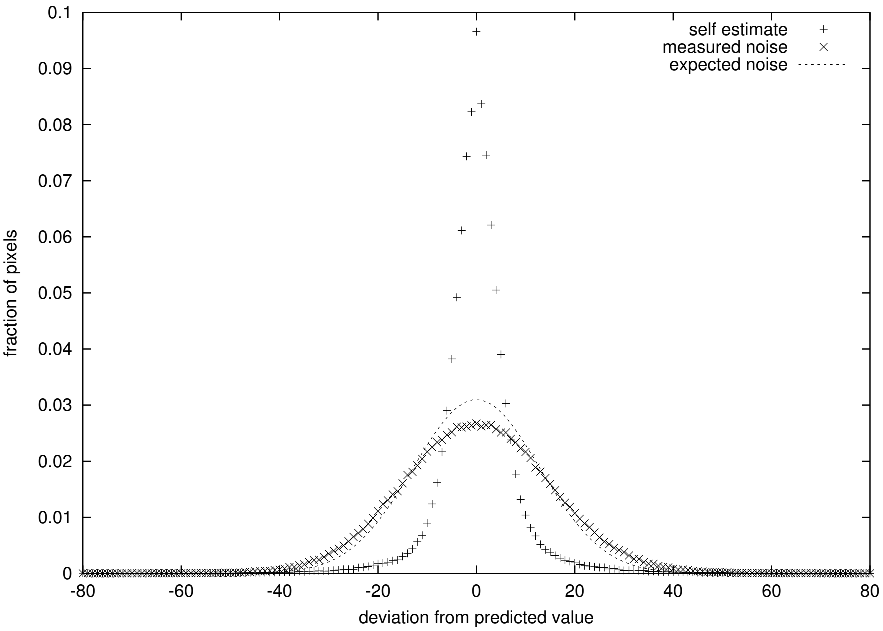


Figure 6.22: Using the model for self-verification and to estimate the strength of unknown noise. The plot shows the distribution of differences between the predicted value and the true pixel in the original image (+) and a noise image (×). The dashed line shows the expected true noise distribution.

6.4 Texture segmentation

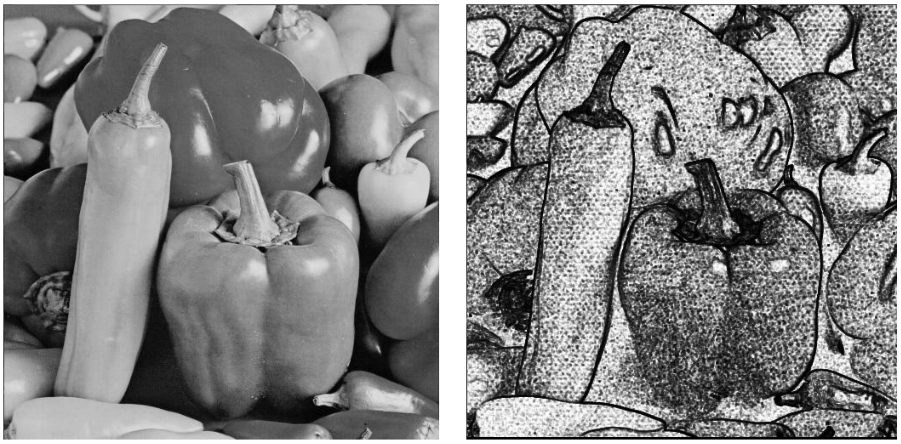
In a way that is similar to the procedure to be introduced in Chapter 7, the Z_q model can be used to extract features from images. We apply the approach described in Section 5.6 to the image shown in Figure 6.23 (top left). This requires learning the bond probabilities for the whole image and then filtering with the resulting probabilistic filter. This means, the resulting image has a value of $\tilde{l}_x^y = 256 \times P(l_x^y | l_i^j, i, j \in \mathcal{N}_{x,y})$ for each pixel, where P is the local probability for the central point $l_{x,y}$ of a 3×3 block. As can be seen, the black pixels in the filtered image (top right) belong to the lines and edges in the image (low probability) and the white pixels to the smoother regions (bulk sample and high probability). This image can now be processed further to separate the objects by using simple histogram thresholding. For comparison the lower left image shows the result of applying a Sobel filter followed by a histogram equalization, and the lower right image is obtained by using the method introduced in Chapter 7.

The histogram of image 6.23 (top right) shown in Figure 6.24 has a local minimum which is used to distinguish between two classes. The corresponding pixels are displayed in the feature image 6.25, where the black pixels belong to the first class (lines, edges) and the white pixels to the bulk sample.

Apart from being useful, the resulting images also have a certain aesthetic value [2], which can be enjoyed in Figure 6.26. The image was first decomposed into its three channels (red, green, blue), the filter was applied to each one and the resulting images recombined to give the final image. Note that the small blocks which appear in the upper part are not introduced by the filter, but are artifacts of the lossy compression scheme (JPEG) the original was subject to. Hence, the procedure can also be used for detecting manipulations in images.

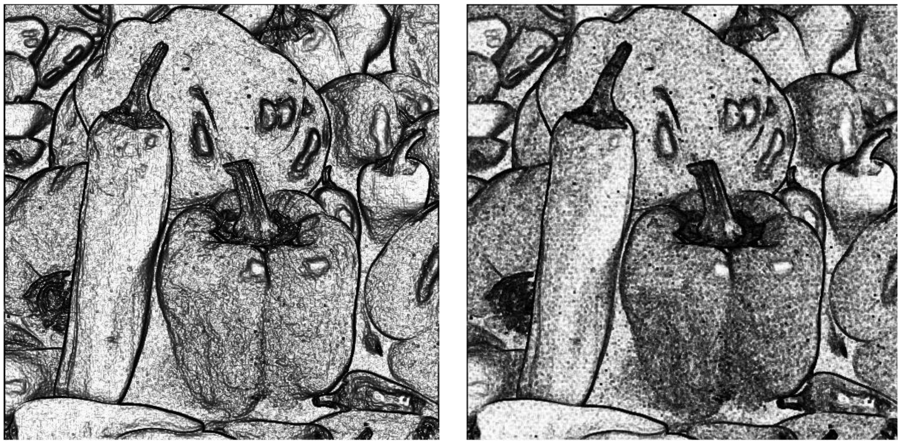
---

[2] As in most cases this is a matter of taste.



original

Zq−block filtered



Sobel filter

ica filtered

Figure 6.23: The Z_q model used for feature extraction. The top left image was filtered with a probability filter consisting of a product of bond probabilities. The top right image is the result of assigning this joint probability to the center pixel (multiplied by 256). The lower two are obtained by conventional methods.

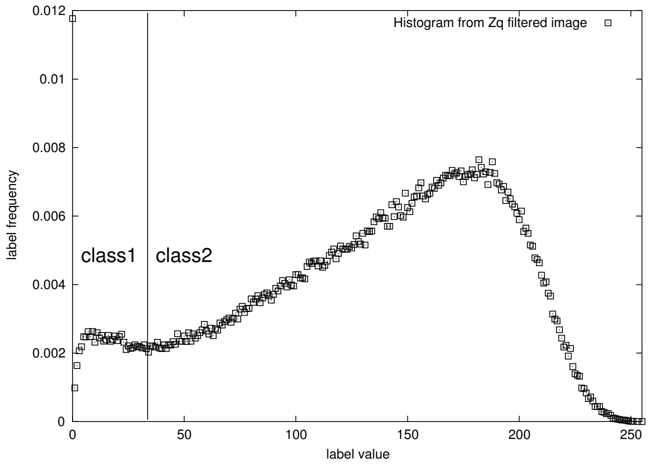


Figure 6.24: Histogram of probability labels from image 6.23 (top right). Two classes can be found from the distribution.

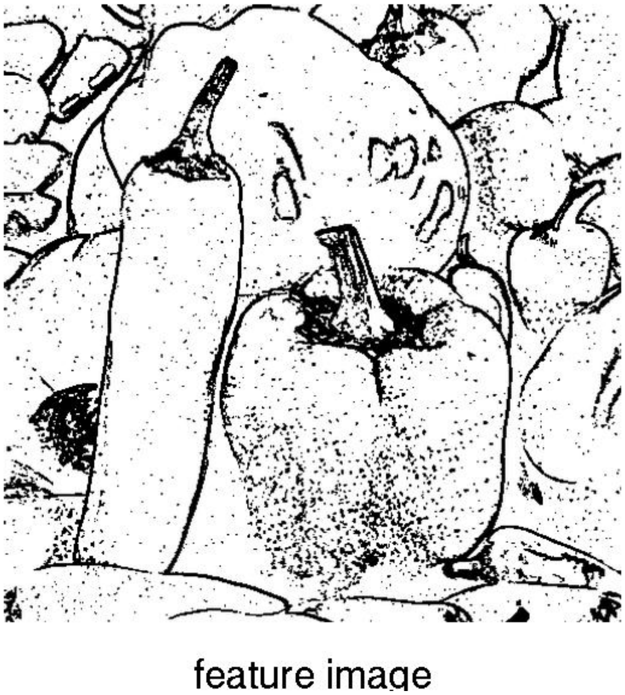


Figure 6.25: Feature image obtained from image 6.23 (top right). The black points correspond to class 1. The white points to class 2. (From Figure 6.24.)

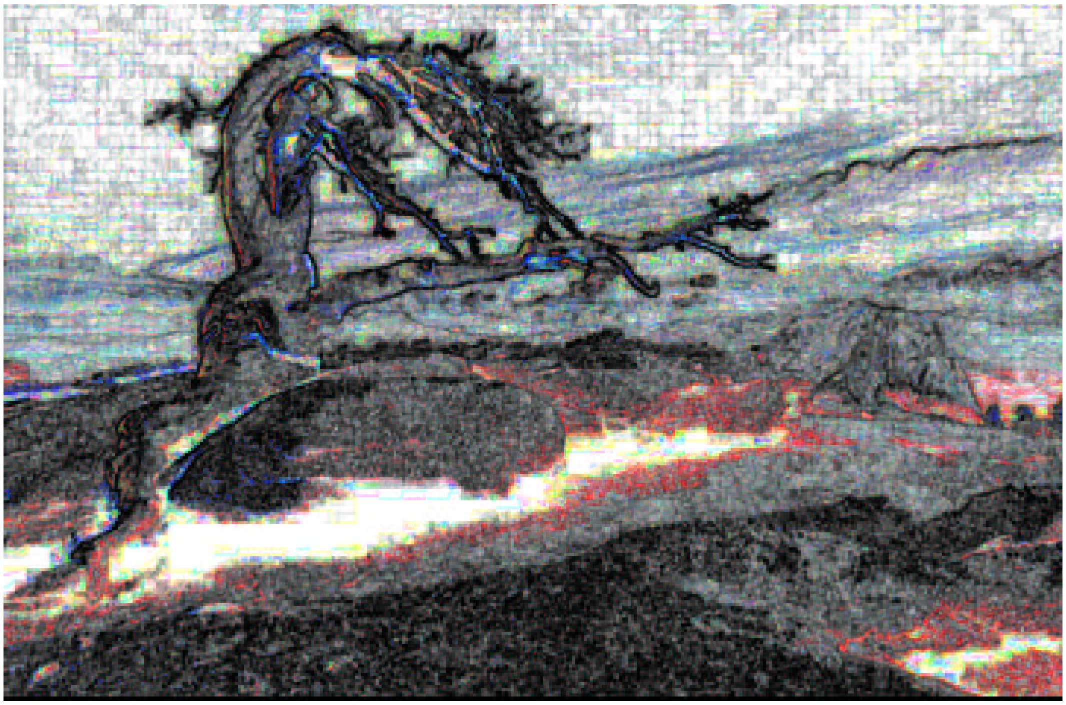


Figure 6.26: The art of feature extraction.

Chapter 7

Feature extraction using Transformed Probabilities

This chapter describes a new technique in which the image or any arbitrary data is transformed into components according to their individual characteristics. In the transformed space the joint probability of a sample (window) can easily be estimated by the product of the marginal probabilities. The resulting probability density can then be used to identify individual clusters corresponding to features in the image.

This chapter does not strictly fit into the context of Markov Random Fields and is for this reason treated separately, including the experimental results.

7.1 Introduction and Motivation

A major step towards a more abstract level of visual processing is the separation and identification of different regions and objects in an image. Our own visual system performs complex tasks to distinguish objects in a scene in order to understand the relation between these objects (or subjects). The main features for isolating objects are: intensity, boundary lines, colour and texture. The low-level processing stages responsible for extracting these features take place very early – in our retina [Hub89]. The abstraction from these features however, happens in the higher paths of our visual system. In order to understand this system a profound analysis of the first stages is necessary.

Apart from these theoretical interests there are also numerous technical applications in which feature extraction is used. To illustrate this, two examples are given here:

Example 1: A bank has to deal with hundreds or thousands of credit transfer forms from their customers per day. Even though online banking is gaining more and more fans, the majority of forms is still filled out by hand. The task is now to automatically separate the hand-written entries from the preprinted form. This can readily be done by a separation in colour space, since transfer forms (at least in Germany) only contain the colours red, yellow and green and the customer's writing is blue or black.[1] The isolated characters and numbers are then further processed by optical character recognition systems (OCRs).

Example 2: In satellite imaging it is often necessary to identify and measure the areas of different terrains. For example the size of a natural forest compared to the progressing clearcut. Usually several different images of the same scene are available, since a satellite contains several imaging systems for different ranges of wave length (multi-band images). Additional information can be drawn from the difference in texture of the terrain. Combining this information enables the analyzing system to automatically measure the regions of interest [Rus99].

The method used here is a member of the class known as 'transformation coding'. Similar methods are incorporated in all state-of-the-art image compression standards like JPEG, JPEG-2000 or MPEG. Note that well known methods like the Fourier-Transformation also belong to this class.

In the following approach the coefficients of the linear transformation are determined by requiring that certain statistical properties of the resulting image hold. These requirements result in a transformation in which the pixels in the image are (blockwise) statistically independent and the joint probability is simply the product of the single pixel probabilities. The result is that every image block can be labeled with its probability meaning high probabilities for the more common blocks (mostly homogeneous regions) and lower probabilities for the less common ones (textural regions and boundaries).

---

[1] If you want to irritate your bank you can fill out your transfer forms with red ink.

7.2 The problem stated

Given a finite set of zero mean random vectors $\vec{s}^k \in \mathbb{R}^N$, where $k = 1, \ldots, M$, with the unknown *exact* probability density $p(\vec{s})$ we aim at approximating this density by a product of marginal probabilities of the transformed vector components $\hat{\vec{s}}$. The reason why we usually can not sample $p(\vec{s})$ itself is that the number of possible points in the sample space is simply too large to give any reasonable statistical estimate. The sample window size in image processing is usually 9 or 25 points, which results in up to 256^{25} configurations. However M is in most applications of the order of 256^2 only.

The central statement is

$$p(\vec{s}) \approx p(\hat{\vec{s}}) = \prod_{i=1}^{N} p_i(\hat{s}_i), \tag{7.1}$$

where the transformation is linear

$$\begin{aligned} \hat{\vec{s}}^k &= \mathbf{A}\vec{s}^k & (7.2) \\ \hat{s_i}^k &= \sum_{j=1}^{N} a_{ij} s_j^k. & (7.3) \end{aligned}$$

$\mathbf{A}$ is a fixed $N \times N$ 'mixing matrix' whose rows $\vec{a_j}$ are the base vectors of the transformation.

A widely used method for calculating this matrix is known as Independent Component Analysis (ICA) [NP94, BS95, Com94, KOW$^+$95, Hyv98]. ICA is a recently developed extension of the standard Principal Component Analysis (PCA). Applications range from image to audio processing, neural networks and unsupervised learning, utilizing mainly the fact that the ICA can be used for blind source separation and feature extraction [HO99].

7.3 Independent Component Analysis

Starting from Equation (7.1) we need to define a measure for independency, since the statement itself is not suitable for finding the matrix $\mathbf{A}$. One general approach [Com94] is based on the concept of mutual information. We define the differential entropy H of the random vectors $\hat{\vec{s}}$ with density $p(\hat{\vec{s}})$ as

$$H(\hat{\vec{s}}) = -\int p(\hat{\vec{s}}) \log(p(\hat{\vec{s}})) d\hat{\vec{s}}. \tag{7.4}$$

The differential entropy can be normalized to provide the definition of negentropy $J(\hat{\vec{s}})$, which has the additional property of being invariant under linear transformations

$$J(\hat{\vec{s}}) = H(\hat{\vec{s}}_{gauss}) - H(\hat{\vec{s}}), \tag{7.5}$$

where $\hat{\vec{s}}_{gauss}$ is a Gaussian random variable of the same covariance matrix as $\hat{\vec{s}}$. In this sense negentropy can be interpreted as a measure of nongaussianity.

With the concept of differential entropy, we can define the mutual information I as a measure of dependence between the random variables $\hat{s_i}, i = 1 \dots N$. Constraining the variables to be *uncorrelated* we have

$$I(\hat{s_1}, \hat{s_2}, \dots, \hat{s_N}) = J(\hat{\vec{s}}) - \sum_i J(\hat{s_i}). \tag{7.6}$$

Since mutual information is the information-theoretic measure of the independence of random variables, we define the ICA of a random vector $\vec{s}$ as the linear transformation $\mathbf{A}$ which *minimizes the mutual information of the transformed components* $\hat{s}_i$ or – recalling that the negentropy is invariant for linear transformations – this is equivalent to *finding directions in which the negentropy is maximized.*

The tasks is now to approximate the negentropy by a suitable function, since the estimation using the definition would require an estimate of the (unknown) probability density function. The classical method of approximating the negentropy is using higher order moments, but it was shown in [Hyv98] that a general form is more suitable

$$J(\hat{s_i}) \approx c \left(\langle G(\hat{s_i}) \rangle - \langle G(\nu) \rangle \right)^2, \tag{7.7}$$

where G is any non-quadratic function, c is an irrelevant constant, and ν is a Gaussian variable of zero mean and unit variance.

Combining these results we end up with the following optimization problem:

$$\begin{aligned} &\text{maximize} \qquad \sum_{i=1}^{N} J_G(\vec{a}_i) \quad \text{for all } \vec{a}_i, i = 1, \dots, N && (7.8) \\ &\text{under constraint} \qquad \langle (\vec{a}_j^T \vec{s})(\vec{a}_k^T \vec{s}) \rangle = \delta_{jk}, && (7.9) \end{aligned}$$

where

$$J_G(\vec{a_i}) = \Big(\langle G(\vec{a_i}^T \vec{s}) \rangle - \langle G(\nu) \rangle \Big)^2 . \tag{7.10}$$

The constraint (7.9) means that the projections of the random vectors onto the different ICA components are uncorrelated.

Prior to solving the optimization problem it is useful to preprocess the data. Since the random vectors are usually not centered the mean vector $\vec{m}$ is subtracted from all members. The second useful preprocessing step is to *whiten* the data. This means, the observed vector $\vec{s}$ is transformed linearly so that the new vector $\tilde{\vec{s}}$ is white, i.e. its components are uncorrelated and their variances equal unity. The most common method for whitening is to use the eigenvalue decomposition of the covariance matrix $\langle \vec{s}\vec{s}^T \rangle$ by singular value decomposition (SVD) [PTVF92].

The method used here for determining the independent components and the matrix $\mathbf{A}$ is based on a fixed-point iteration scheme (FastICA) introduced in [Hyv98]. To find the maxima of Equation (7.9) it starts from the Lagrange-Function (for one component)

$$L(\vec{a_i}) = \langle J_G(\vec{a_i}) \rangle - \lambda \langle \vec{a_i}^T \vec{a_i} \rangle \tag{7.11}$$

and according to the Kuhn-Tucker conditions the optima of $\langle J_G(\vec{a_i}) \rangle$ are obtained at points where

$$\langle \vec{s}\, g(\vec{a_i}^T \vec{s}) \rangle - \lambda \vec{a_i}^T = 0, \tag{7.12}$$

where $g(x) = G'(x)$. The FastICA scheme solves the optimization problem by a simple iterative procedure, based on Newton's method.

7.4 Estimation of the probability density function

The above procedure returns the components of the matrix $\mathbf{A}$. All common applications use only the *vectors* as a base for the data samples. The new idea is to calculate the marginal probability density function $p(\hat{s}_i)$ for each transformed component $\hat{s}_i = \vec{a_i}^T \vec{s}$ by sampling the projected values in a histogram and estimating the one-dimensional probabilities from these frequencies. Note that this requires a quantization of the values, since $\hat{s}_i \in \mathbb{R}$. Multiplcation finally gives the *estimated joint probability* for the random vectors as stated in Equation (7.1). The advantage of this method is that we obtain an approximation to the true probability which can not be sampled, since the dimension of the samples is simply too large. The resulting probability can then be calculated for each sample again to tell us with what probability the sample belongs to the bulk sample. The interesting members will deviate from this and occur with a lower probability. This can be utilized to isolate these features.

7.5 Labeling the data

We can now assign a probability value to each single input/source vector $\vec{s}$ which is in this case a sample window taken from an image. Each component s_i is the pixel value corresponding to the intensity of the image inside a specified sample window. A simple procedure visits each pixel $s_{x,y}$ in the image, calculates the probability of the surrounding window (taking this pixel as the center point) and then setting the pixel value in the 'new' image to the value of $p(s_{x,y})$. In order to gain a better visual impression the new value can be scaled linearly or non-linearly by any positive function. From this new image we can calculate the histogram of image intensities (equal to the histogram of probabilities). If there are in fact regions containing differing textures in the original image, these will show up in the histogram as different peaks, which are then easy to separate. This can be done automatically or by hand. One disadvantage in the preceding scheme is the fact that the mean vector is subtracted from the data and the total (offset) intensity is removed from the data. A simple solution to this is to create a two-dimensional histogram with the local mean (or single pixel) intensity plotted vs. the intensity in the image $p(s_{x,y})$. This can, in principal, be expanded for every new feature extracted from the image.

There are other methods known to the image processing community. Most of them use the fact that textures have different local correlations between pixel values. The neighbourhood of a pixel is combined to a single value which serves as a new label for this point. Several methods simply use the local standard deviation or apply one of various filters summing up the surrounding pixels multiplied by the corresponding filter coefficient. The combined histogram is then used to isolate different clusters belonging to different textures (for a complete overview see [Rus99]).

7.6 The whole program

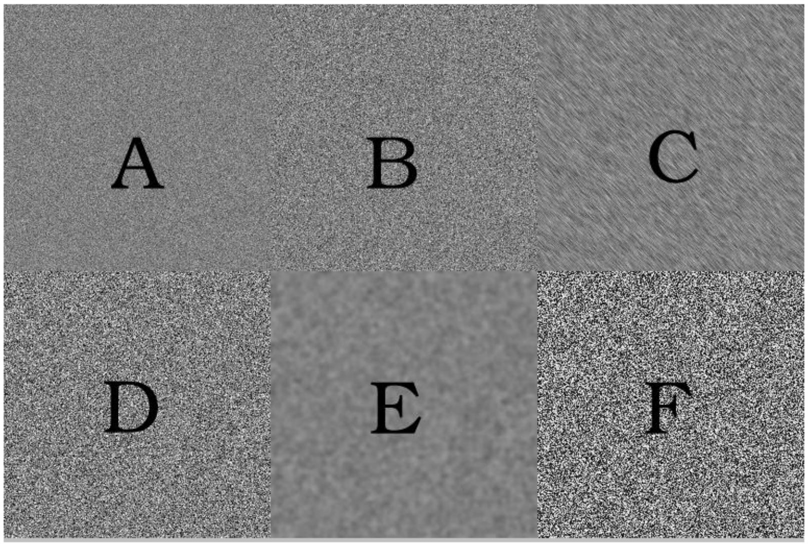
In order to make it easier to follow the steps, a complete list is given here again:

1. Subtract mean value $\langle\vec{s}\rangle$ from data $\vec{s}_k, k = 1, \dots M$
2. Calculate covariance matrix $Cov(\vec{s})$
3. Determine (normalized) eigenvectors and eigenvalues λ_i of $Cov(\vec{s})$ (PCA)
4. Project data onto eigenvectors and normalize with $\sqrt{\lambda_i}$ (whitening), project back
5. Do ICA to find mixing matrix $\mathbf{A}$, using $g(x) = G'(x) = tanh(bx), b \in [0, 1]$
6. Transform data using $\hat{\vec{s}} = \mathbf{A}\vec{s}$
7. Calculate probability density $p(\vec{s_i})$ for each component $\hat{s}_i$ (needs quantization)
8. Calculate joint probability p for each $\hat{\vec{s_i}}$ and assign this p to this data sample

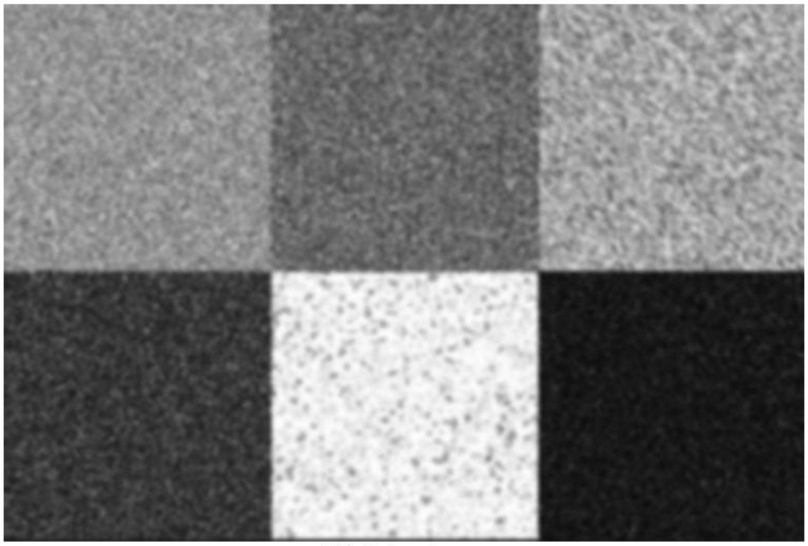
   Further processing:
9. Label each data sample with an integral number ξ (0 to 255), where $\xi = \xi(p)$ (for example $\lceil 255 \times p \rceil$)
10. Calculate combined 2-dimensional histogram of s_i and ξ_i
11. Identify different features by separating single clusters in the histogram

7.7 Applications

The following sections contain two example applications for the feature extraction. The first is a simple phantom test image to illustrate and evaluate the procedure, the second is a ‘real-life’ application.



original image



ICA feature image

Figure 7.1: Set of phantom test images. The upper image contains 6 regions of different textures: A) Gaussian $\sigma = 20$, B) Gaussian $\sigma = 30$, C) Gaussian $\sigma = 40$ and blurred by a diagonal 5x1 filter, D) Gaussian $\sigma = 50$, E) Gaussian $\sigma = 50$ and blurred by a Gaussian filter with $r = 7$, F) uniform in the range 0 to 255. All images have a mean intensity of 127. The lower image displays the pixels labeled with the corresponding joint probability (from ICA).

7.7.1 Phantom test image

A test set of 6 different textures is used to verify the feature extraction. The image in Figure 7.1 is constructed artificially by assigning each pixel an intensity drawn from different random distributions. These are

A Gaussian distribution with $\sigma = 20, \mu = 127$

B Gaussian distribution with $\sigma = 30, \mu = 127$

C Gaussian distribution with $\sigma = 40, \mu = 127$, blurred afterwards with a 5x1 diagonal filter

D Gaussian distribution with $\sigma = 50, \mu = 127$

E Gaussian distribution with $\sigma = 50, \mu = 127$, blurred afterwards with a Gaussian filter $r = 7$

F uniform distribution, values in the range [0..255]

All textures have the same mean intensity value. This means that the textures can not be separated using the pixel intensity histogram, since all of the single histograms overlap.

After applying the feature extraction method – which results in the lower image – all regions have different and non-overlapping intensity distributions. Combining the original pixel intensity information with the pixel labels results in the two-dimensional histogram in Figure 7.2. The plot exhibits 5 distinct peaks which correspond to the different regions. However, the regions A and C merge together in one cluster and are not easily separated. A possible way to subdivide this peak is to run the feature extraction again, this time only for the two regions. This can in principle be done as a kind of iterative method to find smaller and smaller clusters.

7.7.2 Isolating the torn edge of a paper

An application that recently has gained interest in the public is the reconstruction of damaged documents. In this case – although maybe not a very common one – the paper sheets containing important information were torn into small pieces. However, the size of the single pieces are still large enough to put them together again in order to recover the original paper [Blu00]. The problem that arises is to find the corresponding 'partner' piece from a larger amount of candidates. One step towards the solution is to use the fact,

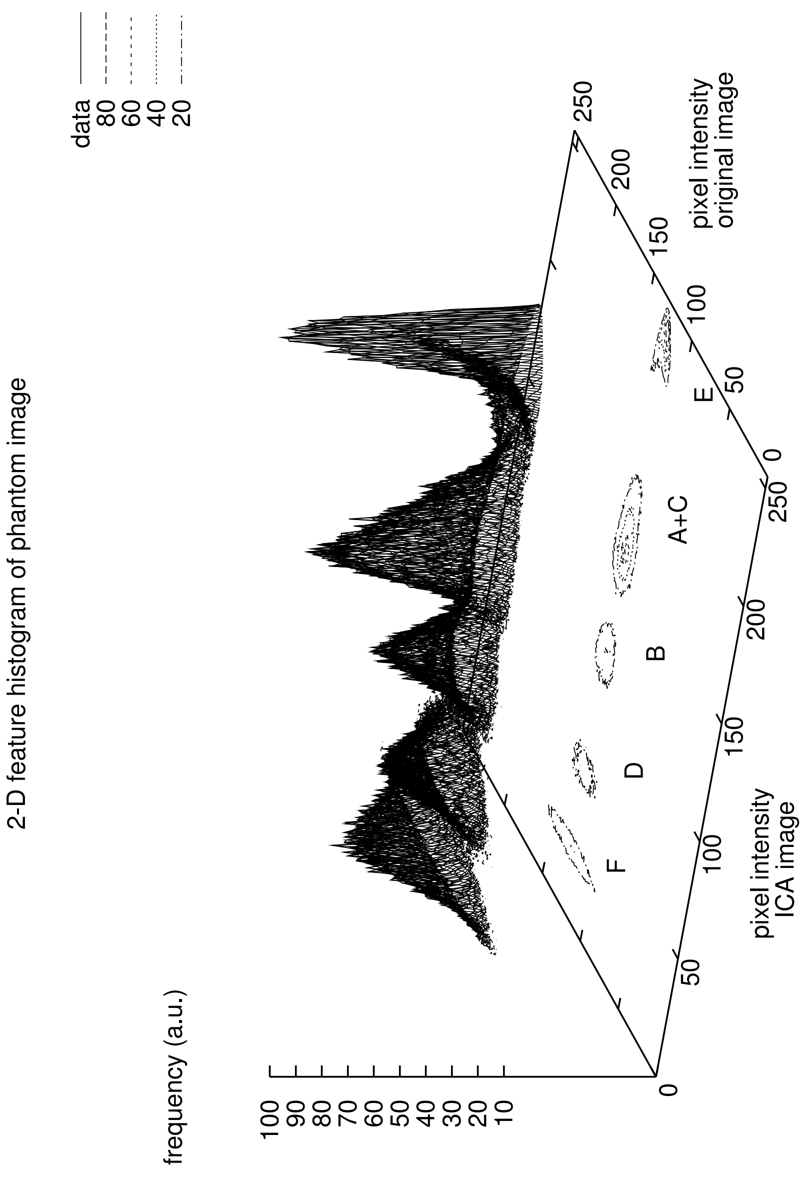


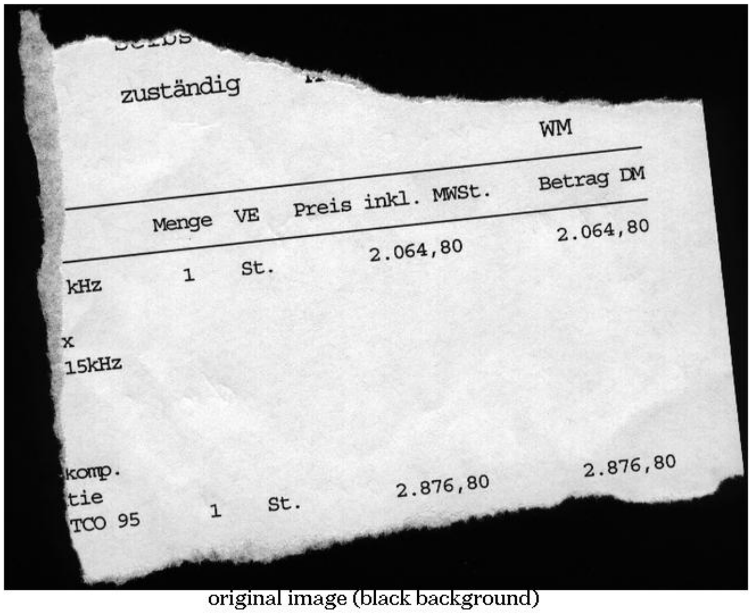
Figure 7.2: Two dimensional histogram combining the images features in Figure 7.1. The projections on the x-y plane contain the points belonging to the different texture regions.

that two piece show a similar edge area where they were torn in two. This can be done automatically by first scanning the paper, isolating the region of interest and then searching for matching partners. Feature extraction becomes necessary for identifying the edge area.

Figure 7.3 shows an example of a piece of paper that was scanned with 256 grey-levels. The area where the paper was torn is visible to a human observer, but the intensity is not homogeneous in this region and hence, no histogram threshold is possible. A texture recognition is necessary.

The first step is to remove any writing, since this will interfere with the detection of textures. A simple way to do this is to scan the object first on a black and then on a white background. The white version is then inverted (the new pixel value is the current intensity subtracted from the maximum intensity) and then added (pixel-wise) to the black background image.

This results in an image with the writing removed from the paper. After applying the labeling procedure with ICA, the black edges are clearly visible (Figure 7.4 – upper image) and are isolated by simple thresholding of the intensity histogram. The extracted regions are shown in Figure 7.4 – lower image.

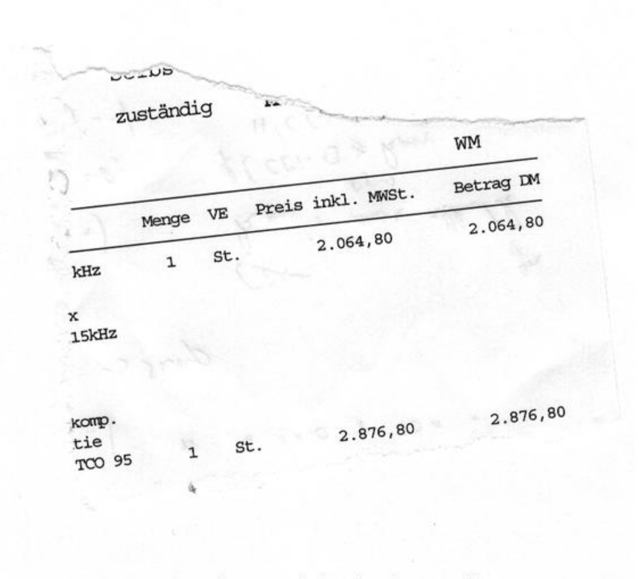


zuständig

WM

	Menge	VE	Preis inkl. MWSt.	Betrag DM
kHz	1	St.	2.064,80	2.064,80
x				
15kHz				
komp.				
tie				
TCO 95	1	St.	2.876,80	2.876,80

original image (black background)



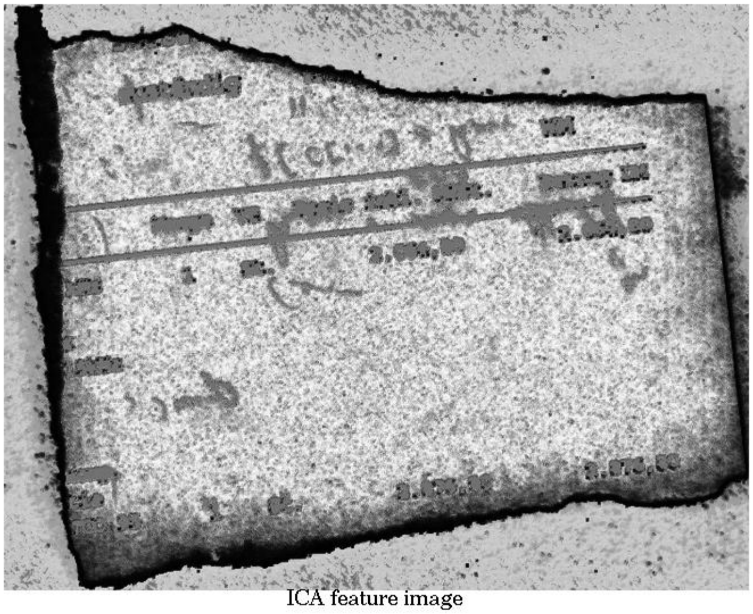
zuständig

WM

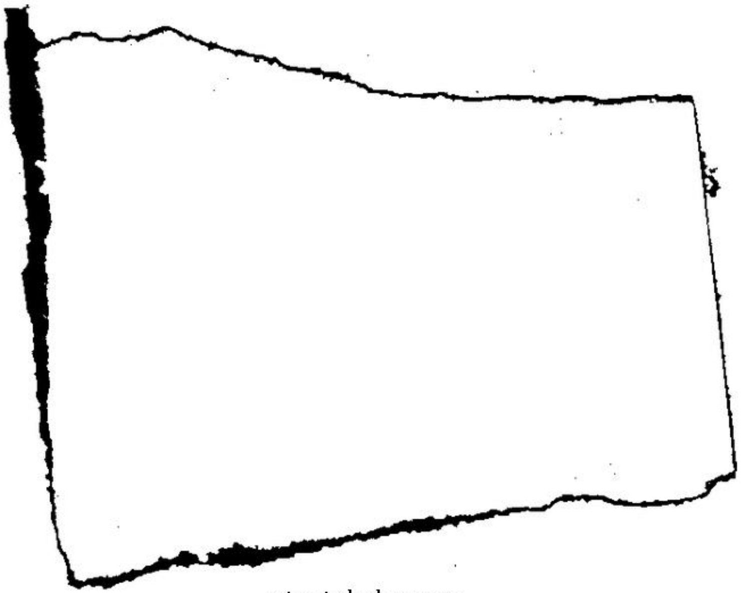
	Menge	VE	Preis inkl. MWSt.	Betrag DM
kHz	1	St.	2.064,80	2.064,80
x				
15kHz				
komp.				
tie				
TCO 95	1	St.	2.876,80	2.876,80

original image (white background)

Figure 7.3: Top: scanned original image of a piece of paper on a black background. Bottom: on white background.



ICA feature image



extracted edge areas

Figure 7.4: Top: ICA processed image. Bottom: final image containing the extracted edge areas (after histogram thresholding and single pixel noise removal).

Chapter 8

Conclusion

This thesis introduced new statistical models for image processing based on the formalism of Gibbs Random Fields. The probabilistic approach describes the interaction of neighbouring pixels by local potentials resulting in a global energy function of the Gibbs distribution. The energy depends on a set of free parameters – the coupling constants – which have to be learned from the data. The learning problem is solved analytically by approximating the true two-dimensional structure of the image by an independent row (or column) approach which decouples into two systems. The image restoration process uses a Bayesian approach based on an extended Monte Carlo procedure. As can be seen from the results even this very simple model, which only includes the interaction between two neighbouring points, is able to capture most of the information contained in the images. This is due to the fact that the main property of natural images – the smoothness – is a very strong constraint for local models. A special property of the Z_q-model is that the energy function is fully parameterized by its 'spectral' components, which means it does not assume any special underlying function of the couplings and hence, can adapt perfectly to the measured correlations. It also means that lines and edges do not have to be treated separately as in the case of most other models. Additionally, the model appears to be very robust under distortion of the coupling constants and learned parameters can even be taken from a noisy image or from another image.

Compared to conventional filters – for example a simple median filter – the restoration performance is very good especially in the very high noise domain, where even our eye is unable to determine the original scene. The model can easily be extended to colour images, by using a metric in the colour space and simply mapping the distance between two colour values to the Z_q states.

Although the approach is based on statistical properties of both the image data and the noisy channel, it can handle deterministic convolutional filters like blur – which are not probabilistic – and still achieve good restoration results. It is shown that even these filters introduce an information loss, which is mainly due to quantization error.

For images where the noise model and especially the parameters are unknown, a predictive procedure can be used to find this model and to estimate the corresponding parameters, by comparing the predicted value of the prior model with the received pixel value.

With these properties the model can in principal be used for lossless compression, since it achieves entropy rates by prediction close to the ones obtained by other state-of-the-art algorithms. However, the model is based on four surrounding neighbours, whereas other models are based on prediction by using other points, mostly from the preceeding point and row.

Another application is image segmentation or classification, where a region of the image is labeled according to its local texture. An approach based on a product of independent bonds utilizing the Z_q model is applied to extract features from an image, segmenting it according to different textures. Another model that aims at capturing the true underlying joint probability of image blocks is applied to a special region in an image in order to separate it from the unimportant data.

From a **physical point of view** the application of statistical mechanical models to image processing enables one to utilize all the known properties of these systems and derive interesting methods which prove to be very powerful and robust. Additionally, the combination of rather theoretical models and real-life data makes this combination an extremely challenging yet satisfying task.

Apart from being interesting, the application in image processing (restoration and compression) is a fundamental problem for data acquisition, transmission, and investigationin today's world.

Finally, as was shown in a few experiments, the model has some interesting connections to **biology**. It appears that low level vision has properties which might have similar processing steps incorporated in the first stages of our visual path. The statistics of natural images – being the input to our brain – play an important role in compression and transmission of visual data from the retina to the visual cortex. Having a simple model of the first stages might help us to understand what's happening in the more complex regions of our brain.

Bibliography

[Abu90] H. Abut. *Vector Quantization.* IEEE Press, 1990.

[ANS96] *ANSI T1.801.01-1996, Digital Transport of Video Teleconferencing / Video Telephony Signals - Video Test Scenes for Subjective and Objective Performance Assessment*, 1996.

[AT43] J. Ashkin and E. Teller. *Physical Review*, 64(178), 1943.

[Ati92] J.J. Atick. Could information theory provide an ecological theory of sensory processing? *Network*, 3:213–251, 1992.

[Bar89] H.B. Barlow. Unsupervised Learning. *Neural Computation*, (1):295–311, 1989.

[Bes74] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36:192–236, 1974.

[Bes86] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.

[BH92] M.F. Barnsley and L.P. Hurd. *Fractal Image Compression.* AK Peters Ltd, Wellesley, Ma, 1992.

[Blu00] W. Blum. Die Schnipseljagd. *Die Zeit*, April 6. 2000.

[BP98] J. Berts and A. Persson. Objective and subjective quality assessment of compressed digital video sequences. Master's thesis, Department of Signals and Systems, Chalmers University of Technology Goeteborg, Sweden, 1998.

[BS95] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, (7):1129–1159, 1995.

[BZYJ96] M. Berthod, Z.Kato, S. Yu, and J.Zerubia. Bayesian image classification using Markov random fields. *Image and Vision Computing*, (14):285–295, 1996.

[CMZ89] G.W. Cottrell, P. Munro, and D. Zipser. Image compression by back propagation. In N.E. Sharky, editor, *Advances in Cognitive Science*. Able, New York, 1989.

[Col99] D.A. Coley. *An introduction to genetic algorithms for scientists and engineers*. World Scientific, 1999.

[Com94] P. Comon. Independent Component Analysis – a new concept? *Signal Processing*, 36:287–314, 1994.

[CT91] M.T. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley-Interscience Publication, 1991.

[Dom74] C. Domb. *Phase Transitions and Critical Phenomena*. Academic Press, 1974.

[DPR81] F. Deák, A. Patkós, and P. Ruján. Thermal characteristics of the two-dimensional planar rotator model. *Physical Review B*, 24(5):2608–2612, September 1981.

[EFF93] A. El-Fallah and G.E. Ford. Image filtering by gradient inverse inhomogeneous diffusion. *Proceedings of the IEEE International Conferens on Acoustics, Speech and Signal Processing*, 5:V73–6, 1993.

[Fie87] D.J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, (4):2379–2394, 1987.

[Fie94] D.J. Field. What is the goal of sensory coding? *Neural Computation*, (6):559–601, 1994.

[Fis95] Y. Fisher. *Fractal Image Compression, Theory and Application*. Springer Verlag, New York, 1995.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Learning*, 6(6):721–741, 1984.

[Gol89] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesly, 1989.

[Gra88] P. Grassberger. Finite Sample Corrections to Entropy and Dimension Estimation. *Physics Letters A*, 128(6.7):369–373, 1988.

[HC71] J.M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. unpublished, 1971.

[HO99] A. Hyvärinen and E. Oja. Independent Component Analysis: A Tutorial. http://www.cis.hut.fi/projects/ica/, 1999.

[Hub89] D. Hubel. *Auge und Gehirn : Neurobiologie des Sehens.* Spektrum der Wissenschaft, Heidelberg, 1989.

[Huf52] D.A. Huffman. *Proceedings of the Institute of Radio Engineers*, 40:1098–1101, 1952.

[Hyv98] A. Hyvärinen. New Approximations of differential entropy for independent component analysis and projection pursuits. *Advances in Neural Information Processing Systems*, 10:273–279, 1998.

[Jac89] A. Jacquin. *A fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding.* PhD thesis, Georgia Institute of Technology, August 1989.

[Jai81] A.K. Jain. Image data compression : A review. *Proc. of the IEEE*, 69(3):349–389, 1981.

[KD92] J. Konrad and E. Dubois. Bayesian Estimation of Motion Vector Fields. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.

[KGV83] S. Kirkpatrick, C.D. Gellat, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[KK96] J. Kangas and T. Kohonen. Developments and applications of the Self-Organizing Map and Related Algorithms. *Mathematics and Computers in Simulation*, 41(5-6), July 1996.

[Kog79] J.B. Kogut. Lattice gauge theory and spin systems. *Reviews of Modern Physics*, 51(4):659–713, October 1979.

[KOW$^+$95] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo. A Class of Neural Networks for Indenpendent Component Analysis. Report A28, Helsinki University of Technology, Finland, October 1995.

[Li95a] S.Z. Li. *Markov Random Field Modeling in Computer Vision.* Computer Science Workbench. Springer Verlag, 1995.

[Li95b] S.Z. Li. On discontinuity Adaptive Regularization. Preprint, 1995.

[MMP87] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 32(397):76–89, March 1987.

[MN79] E. Meyer and E. Neumann. *Physikalische und Technische Akustik.* Vieweg, Braunschweig, 1979.

[MPG] *ISO/IEC IS 13818. Information Technology – Generic Coding of Moving Pictures and Associated Audio Information.*

[MRR$^+$53] N. Metropolis, A.W Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by a fast computational machine. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[NK88] N.M. Nasrabadi and R.A. King. Image coding using vector quantization : A review. *IEEE Trans. on Communication*, 36(8):957–971, 1988.

[NP94] J.P. Nadal and N. Parga. Non-linear neurons in the low noise limit: A factorial code maximizes information transfer. *Network*, (5):565–581, 1994.

[PB94] M. Potters and W. Bialek. Statistical Mechanics and Visual Signal Processing. *J. Phys. I France*, (4):1755–1775, 1994.

[PB95] J.M. Pryce and A.D. Bruce. Statistical mechanics of image restoration. *Journal of Physics A*, 28:511–532, 1995.

[Pen90] W. Pennebaker. JPEG Technical Specification, Revision 8. Working Document No. JTC1/SC2/WG10/JPEG-150, August 1990.

[PTK85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[PTVF92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992.

[RJ88] V. Ramamoorthy and N.S. Jayant. High Quality Image Coding with a Model-Testing Vector Quantizer and a Human Visual System Model. *IEEE Conf. Acoust., Speech, Signal Processing*, 2:1164–1167, April 1988.

[RM90] S.J. Reeves and R.M. Mersereau. Regularization parameter estimation for iterated image restoration in a weighted Hilbert space. *ICASSP*, 4:1885–1888, 1990.

[Ruj79] P. Ruján. Variational method for lattice systems : General formalism and application to the two-dimensional Ising model in an external field. *Physica A*, 96A:379–412, 1979.

[Ruj93] P. Ruján. Finite Temperature Error-Correcting Codes. *Phys. Rev. Letters*, (70):2968–2971, 1993.

[Ruj00] P. Ruján. private communication, 2000.

[Rus99] J.C. Russ. *The image processing handbook.* CRC Press and Springer Verlag, 1999.

[RWFF81] P. Ruján, G.O. Williams, H.L. Frisch, and G. Forgács. Phase diagrams of two-dimensional Zq models. *Physical Review B*, 23(3):1362–1370, February 1981.

[Sav80] R. Savit. Duality in field theory and statistical systems. *Reviews of Modern Physics*, 52(2):453–487, April 1980.

[Say00] K. Sayood. *Introduction to Data Compression.* Academic Press, Morgan Kaufmann, 2000.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. Journal*, 27:379–423, 623–656, 1948.

[SJ72] C.F. Stromeyer and B. Julesz. Spatial Frequency Masking in Vision: Critical Bands and Spread of Marking. *Journal of the Opt. Soc. of America*, 62(10):1221–1232, October 1972.

[Sou94] N. Sourlas. Spin Glasses, Error-Correcting Codes and Finite-Temperature Decoding. *Europhys Letters*, (25):159–164, 1994.

[WCVG99] T. Wanschura, D.A. Coley, W. Vennart, and S. Gandy. Automatic Realignment of Time-Separated MR Images by Genetic Algorithm. *Magnetic Resonance Imaging*, 17(2):313–317, 1999.

[Wil84] G.O. Wiliams. Parallel Processing: the Ising model and Monte Carlo dynamics. *J.Phys A*, 18:1781–1794, 1984.

[WM96] X. Wu and N.D. Memon. CALIC – A Context Based Adaptive Lossless Image Coding Scheme. *IEEE Transactions on Communications*, (45):437–444, 1996.

Lebenslauf

Name	Thorsten Wanschura	
Geburtsdatum/-ort	13.01.1971 in Oldenburg	
Nationalitaet	deutsch	
Familienstand	ledig	
Schulbildung	77-81	Grundschule, Oldenburg
	81-83	Orientierungsstufe, Oldenburg
	83-90	Altes Gymnasium Oldenburg
Schulabschluss	90	Abitur
Wehrdienst	90-91	in Oldenburg
Studium	91-93	Grundstudium Diplom Physik mit Nebenfach Informatik,Universitaet Oldenburg
	93	Vordiplom
	93-96	Hauptstudium Diplom Physik mit Nebenfach Informatik, Universitaet Oldenburg, Abschluss Diplom
Auslandsaufenthalte	94-95	Studienjahr an der University of Exeter, Grossbritannien; Studienarbeit
	96	Aufenthalt an der Helsinki University of Technology am Laboratory of Computer and Information Science

Oldenburg, den 10.6.2001

Erklärung

Hiermit versichere ich, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Hilfsmittel verwendet habe.

(Thorsten Wanschura)