Learning generative models of mid-level structure in natural images

Nicolas Heess

Doctor of Philosophy

Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh

2011

Abstract

Natural images arise from complicated processes involving many factors of variation. They reflect the wealth of shapes and appearances of objects in our three-dimensional world, but they are also affected by factors such as distortions due to perspective, occlusions, and illumination, giving rise to structure with regularities at many different levels. Prior knowledge about these regularities and suitable representations that allow efficient reasoning about the properties of a visual scene are important for many image processing and computer vision tasks. This thesis focuses on models of image structure at intermediate levels of complexity as required, for instance, for image inpainting or segmentation. It aims at developing generative, probabilistic models of this kind of structure, and, in particular, at devising strategies for learning such models in a largely unsupervised manner from data.

One hallmark of natural images is that they can often be decomposed into regions with very different visual characteristics. The main approach of this thesis is therefore to represent images in terms of regions that are characterized by their shapes and appearances, and an image is then composed from many such regions. We explore approaches to learn about the appearance of regions, to learn about region shapes, and ways to combine several regions to form a full image. To achieve this goal, we make use of some ideas for unsupervised learning developed in the literature on models of low-level image structure and in the "deep learning" literature. These models are used as building blocks of more structured model formulations that incorporate additional prior knowledge of how images are formed.

The thesis makes the following contributions: Firstly, we investigate a popular, MRF based prior of natural image structure, the Field-of Experts, with respect to its ability to model image textures, and propose an extended formulation that is considerably more successful at this task. This formulation gives rise to a fully parametric, translation-invariant probabilistic generative model of image textures. We illustrate how this model can be used as a component of a more comprehensive model of images comprising multiple textured regions. Secondly, we develop a model of region shape. This work is an extension of the "Masked Restricted Boltzmann Machine" proposed by Le Roux et al. (2011) and it allows explicit reasoning about the independent shapes and relative depths of occluding objects. We develop an inference and unsupervised learning scheme and demonstrate how this shape model, in combination with the masked RBM gives rise to a good model of natural image patches. Finally, we demonstrate

how this model of region shape can be extended to model shapes in large images. The result is a generative model of large images which are formed by composition from many small, partially overlapping and occluding objects.

Acknowledgements

I am deeply indebted to my supervisor Chris Williams, whom I had the privilege to work with. Not only have I benefited so much from his extreme breadth and depth of knowledge, his scientific rigor, and his willingness to get to the bottom of things, at least some of which, I hope, have been transferred onto me. He has also given me very large amounts of help, support, and encouragement, and has shown sometimes sheer infinite patience discussing big and small problems and ideas, giving me a lot of freedom without ever leaving me without advice and guidance.

I would also like to thank John Winn and Nicolas Le Roux for giving me the opportunity to come and repeatedly return to Microsoft Research in Cambridge, where I have spent almost one year in total over the course of my PhD. Joining their project was an extremely exciting and enriching experience. I have learned a lot from both of them and greatly benefited from their ideas and insights, and their help and support during my time in Cambridge and beyond.

There are many other people that I had the opportunity and pleasure to interact with over the course of my PhD, with whom I had inspiring discussions, and who provided me with help and ideas: Amos Storkey, Iain Murray, David Reichert, Kian-Ming Chai, Athina Spiliopoulou, Geoffrey Hinton, Charles Sutton, Guido Sanguinetti, Jyri Kivinen, Andrew Dai, Peter Orchard, and other members of the machine learning group and the Neuroinformatics DTC in Edinburgh as well as in the MLP group in Cambridge. Especially I would like to thank Hannes Saal and Ali Eslami for proof reading parts of this thesis, and Hannes also for his help with printing and binding.

Finally, I am deeply grateful to my parents Marina and Manfred and my sister Katja. Without their support and encouragement over the whole course of my scientific career and especially in the last stages of the PhD this thesis would have not been possible. And, of course, to Gabby, for her love, her never-ending cheerfulness, her understanding – in short, for making my time in Edinburgh so much more enjoyable.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Nicolas Heess*)

Table of Contents

Chapter 1

Introduction

1.1 Modeling image structure

Modeling the structure in natural images is a challenging problem. Two-dimensional images arise from complicated processes involving many factors of variation at different levels. Even apparently "simple" images which are dominated by a few high-level causes (such as a small number of objects) are highly complex. Not only do they reflect the wealth of shapes and appearances of objects in our 3D world, but they are also affected by factors such as distortions due to perspective, occlusions, and changes in illumination. Nevertheless, despite this enormous variability, natural images exhibit striking regularities. They are normally easily distinguished from other kinds of images, e.g. from artificially generated noise images. Furthermore, a human observer looking at a natural scene or image does not usually perceive simply a large number of incoherent colors but rather has a highly structured perception that reflects important properties of the underlying physical scene in an often surprisingly accurate manner. While the processes leading to this percept are still poorly understood it is generally accepted that it cannot be formed based purely on the evidence available from the light falling on the retina, since this "inverse problem" is ill-posed (e.g. Poggio et al., 1985; see also Horn, 1977). Instead it requires an internal model that allows combination of this evidence with prior knowledge of properties of the visual world. If such an internal model is essential for human perception, so too is it for computer vision and image processing. Many attempts have been made to capture the structure of natural images in probabilistic (and other) models. Yet, despite considerable progress most models or computer vision systems still account only for very specific aspects of images or are designed to solve isolated tasks (such as object detection).

One important property of the structure in natural images is that it can be characterized and interpreted at many different levels. Considering a typical image it is usually easy to distinguish the high-level objects that make up the scene and their spatial relations, and one can identify different parts and subparts of these objects and their arrangements and appearances. At intermediate levels of abstraction natural images can still be parsed reliably into coherent regions with distinct visual characteristics in terms of the shape of their boundaries and of their appearances ("texture"). Spatially separate image elements might be grouped together based on various criteria to form a single perceptual entity without them necessarily having to be recognized as a particular object. Particular types of junctions, shading, or texture gradients can, for instance, give an indication of relative depth or of the shape of surfaces. And even with respect to their most generic properties images exhibit striking regularities such as the highly non-Gaussian and scale invariant statistics of the responses of linear filters.

1.2 Generative vs. discriminative models

One premise of the work in this thesis is that generative models hold important advantages in computer vision. Their perhaps most obvious advantage over discriminative methods is that they are more amenable to unsupervised learning, which seems of crucial importance in a domain where labeled training data is often expensive while unlabeled data is nowadays easy to obtain. Equally important, however, is that in vision we are rarely interested in solving a single "task" such as object classification in isolation. Instead we typically need to extract information about different aspects of an image and at different levels of abstraction and scale, e.g. recognizing whether an object is present, identifying its position and pose and those of its parts, and separating pixels belonging to the object from those that are part of the background or of occluding objects (segmentation). Many lower-level tasks, such as segmentation, are not even well defined without reference to more abstract structure (e.g. the object or part to be segmented), and information in natural images, especially when it is low-level and local, is often highly ambiguous. These considerations strongly suggest that we need a model that is able to represent and learn a rich prior of image structure at many different levels of abstraction, and that also allows efficient combination of information bottom-up (from the data) and top-down (from the prior) during inference. Probabilistic, generative models naturally offer the appropriate framework for doing such inference. In particular, unlike their discriminative counterparts, they are trained

not with respect to a particular, task-specific, label (which in most cases provides very little information about the complex structure present in an image) but rather to represent the data efficiently. This makes it much more likely that the required rich prior can ultimately be learned, especially if a suitable, e.g. hierarchical model structure is assumed.

1.3 Generative models of mid-level structure

The work described in this thesis aims at developing generative models of natural image structure. Although the long-term goal is to fully capture the structure in natural images, the work that will be presented here focuses especially on modeling structure at an intermediate level of complexity, e.g. related to a decomposition of an image into coherent regions, the grouping of spatially disjunct elements such as regions or contours into larger perceptual entities, the inference of local image depth and occlusions (figure-ground organization), or of three-dimensional shape of surfaces from their shading.

Although high-level causes of a scene are often the primary targets of interest (e.g. in object detection tasks), being able to reason explicitly about such image structure of intermediate complexity is important for at least two reasons: Firstly, it can provide a more parsimonious image representation than the pixel image itself, thus forming a useful foundation e.g. for models of high-level structure. Secondly, this level of representation can provide important information about the organization of a scene and the objects contained in it. For instance, a representation of a image in terms of region texture and shape is the representation that is required for many image processing tasks, such as image segmentation, or inpainting on the scale considered by Bertalmio et al. (2003) and Criminisi et al. (2004), who deal with the infilling of relatively large image regions, e.g. after the removal of a foreground object as is illustrated in Figure 1.1.

Intermediate processing stages and representations have received significant attention in the literature on human perception and biological vision (e.g. Marr, 1977; Nakayama et al., 1995; Palmer, 1999), and also in the computer vision literature (see for, instance, the work by Ren (2006) for a probabilistic but predominantly *discriminative* treatment of mid-level vision), but to a much lesser extent in the literature on generative image models; in this line of work a focus on structure of intermediate complexity is relatively rare (but see e.g. Guo et al., 2003; Tu and Zhu, 2006; Guo et al., 2007). Most generative work to date focuses on either high- or low-level structure. Es-

(a) (b)

Figure 1.1: Example of image inpainting from Criminisi et al. (2004). This task can, in principle, be performed without knowledge of high-level structure such as objects. Instead, the problem at hand requires knowledge about image regions, textures, and boundary shapes in order to fill the region occupied by the foreground object in a plausible manner.

pecially in the computer vision literature, considerable effort has been made to model the high-level structure of objects and object categories, often in a hierarchical fashion (see Felzenszwalb and Huttenlocher, 2005; Jin and Geman, 2006; Epshtein and Ullman, 2007; Fidler and Leonardis, 2007; Ommer and Buhmann, 2010; Zhu et al., 2008; Todorovic and Ahuja, 2008; Bouchard and Triggs, 2005; Zhu and Mumford, 2006, for some examples). This work, however, is often primarily aimed at capturing the structure of specific object categories to an extent necessary for a particular vision task, such as recognition or segmentation, and less on providing a description of general properties of natural images. In most cases it is not possible to generate new images from the learned models, and unsupervised learning can be a problem (but see, for instance, Fidler and Leonardis, 2007; Todorovic and Ahuja, 2008). Other models in the computer vision literature can extract information about shape and appearance, illumination, occlusion and other factors of variation in an unsupervised manner (Frey and Jojic, 2003; Williams and Titsias, 2004; Kannan et al., 2005; Winn and Jojic, 2005; Kannan et al., 2007). Although these models have successfully been applied to sets of relatively homogeneous images, e.g. to images of particular object classes or to movies featuring a small number of objects, they have limited scope and are typically not suitable for more heterogeneous data, let alone natural images in their generality.

At the other end of the spectrum much effort has been devoted to generative models of low level structure which capture fundamental image statistics and can be applied to tasks such as image denoising or simple inpainting problems. Most of these models also allow drawing samples from the learned distribution i.e. generating new images which reflect the kind of natural image structure captured by the models (e.g. Olshausen and Field, 1997; Bell and Sejnowski, 1997; Lewicki and Olshausen, 1999; Hyvärinen and Hoyer, 2000; Roth and Black, 2005; Osindero and Hinton, 2008; Lee et al., 2009; Karklin and Lewicki, 2009; Sinz et al., 2010; Ranzato et al., 2010a,b). These models are typically trained in an unsupervised manner and, unlike many models in computer vision, they make very few and general assumptions about the image formation process and the nature of the representation to be learned. As a consequence, these models are generic in the sense that they can learn about fundamental properties of natural images in their generality, but they are also very limited in their representational capabilities. When trained on natural images they are only able to capture basic image properties such as piece-wise smoothness but fail to account even for structure at intermediate complexity such as regions and region boundaries. Furthermore, for computational reasons, many of these models are limited to small image patches.

One class of models (not completely distinct from the models described in the previous paragraph) that has recently received particular attention and that has raised hopes with respect to the possibility of learning image structure at different levels of complexity are models from the *deep learning* literature (Hinton et al., 2006b; Bengio, 2009) which emphasizes the power of hierarchical, distributed representations in the context of AI-style tasks such as vision (for applications to image modeling see Osindero and Hinton, 2008; Lee et al., 2009; Ranzato et al., 2010a,b). There has recently been some progress towards modeling richer image structure using models from this class, (e.g. Ranzato et al., 2010a,b, 2011) but the question of how to formulate and learn a good generic image prior remains an open problem.

1.4 Outline of the thesis

In order to overcome some of the limitations of generic image priors we will attempt to develop richer models of natural images by employing more structured model formulations that incorporate some additional prior knowledge about the properties of natural images. One hallmark of natural images, which is also reflected by the example in Figure 1.1, is that they can often be decomposed into regions with very different vi-

sual characteristics. The main approach of this thesis is therefore to represent images in terms of such regions. Each region is characterized in terms of its shape and its appearance, and an image is then composed from many such regions. Simply speaking, this requires models of region appearance, of region shape, and an approach to combine these complementary models to form a full image. This thesis explores approaches to learn about the appearance of regions, to learn about region shapes, and ways to combine several regions to form a full image. In particular, one approach that we pursue for combining multiple regions takes into account how natural images are formed and explicitly accounts for occlusion of overlapping image elements. To achieve this goal, the work in this thesis uses some of the ideas for unsupervised learning developed in the literature on low-level image structure and in the "deep learning" literature (as discussed towards the end of the previous section), but combines them with more structured model formulations which are more common in computer vision.

The presentation of this work is structured as follows:

- **Chapter 2** provides some technical background on models used in the later chapters of the thesis.

- **Chapter 3** mainly deals with region *appearance*. It has two goals: It (a) investigates the generative capabilities of a popular probabilistic model of *generic* image structure and it (b) develops a translation-invariant, probabilistic, fully parametric model of image texture, i.e. a model that can be used to model region *appearance*. It further provides an illustration of how this model can be used as a component of a more comprehensive model that composes full images from several regions with different textures. This work has been published as follows:

  - N. Heess, C. K. I. Williams, and G. E. Hinton. Learning Generative Texture Models with extended Fields-of-Experts. In *Proceedings of the British Machine Vision Conference, BMVC 2009, London, UK*. British Machine Vision Association, 2009.

- **Chapter 4** deals with region *shape*: It builds on previous work by collaborators on the Masked RBM (Le Roux et al., 2011), which introduces an explicit notion of shape and appearance into a deep learning framework, and develops an "occlusion-aware" model for the shape of regions in image patches. The full model assumes that an image patch is composed from a set of independent regions each modeled in terms of its shape and appearance. Regions are associated with a relative depth and compose in an occluding manner. The chapter shows

experiments which demonstrate that this gives rise to a viable model of natural image patches.

- **Chapter 5** demonstrates how the model developed in chapter 4 can be used to define a generative model of full images which represents images in terms of many small, partially overlapping regions each of which is associated with a shape and appearance. We demonstrate how this model, when trained on natural images, can be applied to simple image processing tasks. We discuss a recursive extension of this model that allows to model long range structure by grouping elementary regions into larger units using a flexible tree-structured hierarchy.

  The work presented in chapters 4 and 5 has been published as follows:

  - N. Le Roux*, N. Heess*, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 23(3):593–650, 2011. *) both authors contributed equally
  - N. Heess, N. Le Roux, and J. Winn. Weakly supervised learning of foreground-background segmentation using masked RBMs. In *Proceedings of the International Conference on Artificial Neural Networks and Machine Learning - ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 9–16. Springer Berlin / Heidelberg, 2011.

- **Chapter 6** summarizes the results presented in this thesis and concludes with a general discussion.

Chapter 2

Background

This chapter introduces the modeling framework that will be used in the remainder of the thesis. As discussed in the previous chapter the long-term goal is to formalize knowledge about the properties of natural images using *generative*, probabilistic models. Such generative models can be formulated in different ways, and different classes of models have been used for this purpose in the literature. In this chapter we will consider several types of models and discuss their advantages, and the associated difficulties. The aim is not to provide an extensive overview of the application of probabilistic models to vision problems. Instead, we will focus on the technical aspects of the different frameworks with a special emphasis on those models that will form the basis of the work presented later in the thesis.

We will first provide some general background on probabilistic models in section 2.1, in particular we will briefly explain the notions of directed and undirected graphical models, inference, and learning (a more comprehensive treatment can be found e.g. in Bishop, 2006). Section 2.2 gives an overview of different types of models in the vision literature and discusses some of these models in more detail. Finally, in section 2.3 we will explain some specific techniques for approximate inference and learning that will be used in this thesis.

2.1 Graphical models, inference, and learning

When discussing probabilistic models it is often convenient to use graphical representations: *Graphical models* are a convenient way to diagrammatically represent important properties of probability distributions, and, in particular, they allow the distinction of two important classes of models, *directed* and *undirected* graphical models.

A graphical model is a graph consisting of a set of nodes (or vertices), representing the random variables of the model, and a set of edges which express probabilistic relationships between the random variables and thereby also define dependencies – or independencies – between them. In directed models (cf. section 2.1.1.1) , also referred to as *Bayesian networks* these edges are directed, indicating an asymmetric relation between the random variables. In undirected models, or *Markov random fields* (MRF; cf. section 2.1.1.2), these connections are undirected and the relations between random variables are symmetric. Although many graphical models are purely directed or undirected, a mixed formulation is also possible and such models are typically referred to as *chain graphs* (cf. section 2.1.1.3). A particular graphical structure does not directly specify a single probabilistic model. Instead it defines the set of probabilistic models that are consistent with the independence properties the graphical structure implies. A graphical model asserts conditional independence relations between the variables involved, which can be exploited, for instance, to develop efficient strategies for inference and learning that will be valid for all models in the set.

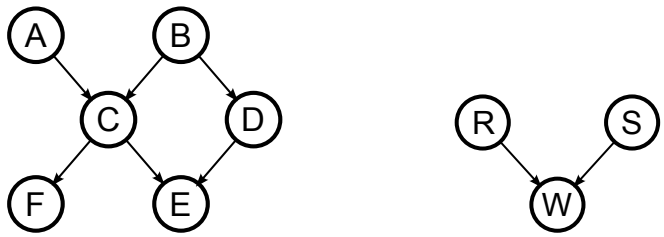
2.1.1 Graphical models

2.1.1.1 Directed graphical models

In *directed* graphical models the edges are drawn as arrows and the graph specifies how a distribution factorizes into conditional distributions. Specifically, the joint distribution that corresponds to a particular graph is given by the product of all conditional distributions associated with the nodes in the graph where the distribution of each node is conditioned on the parents of that node. The joint distribution over the set of variables $\mathbf{x} = \{x_1, x_2, \ldots x_N\}$ is specified by a graph with N nodes (one for each random variable), and it decomposes into a product of local conditional distributions

$$p(\mathbf{x};\Theta) = \prod_{n=1}^{N} p_n(x_n|\mathrm{pa}_n;\theta_n), \tag{2.1}$$

where pa_n is the set of parents of the node x_n in the graph, θ_n the parameters of p_n, and Θ is the set of all parameters. This is illustrated in Fig. 2.1(a). A valid directed graphical model must not contain any (directed) cycles, i.e. the graph must be a directed acyclic graph (DAG). One appealing property of directed graphical models is that they directly specify a procedure to generate samples from the model by taking advantage of the ordering that arises from the graphical structure (this is also referred to as *ancestral sampling*). As a downside inference in directed models, even in seemingly simple



(a) (b)

Figure 2.1: *(a)* Example of a DAG specifying the conditional independencies of a class of distributions. The DAG specifies that the joint distribution factorizes as $P(A,B,C,D,E,F) = P(A)P(B)P(C|A,B)P(D|B)P(E|C,D)P(F|C)$. *(b)* Explaining away in directed graphical models: R and S are marginally independent, i.e. $P(R,S) = P(R)P(S)$, but observing W introduces a dependence between R and S and $P(R,S|W) = P(R)P(S)P(W|R,S)/P(W)$ does not generally factor into $P(S|W)P(R|W)$.

ones, is often hard. This is due to their non-trivial conditional independence properties (see e.g. Bishop, 2006, chapter 8.2 for details) and, in particular, due to a phenomenon referred to as "explaining away": as illustrated in Fig. 2.1 (b) observing a variable that is a child-node in a DAG can introduce dependencies between marginally independent parents. Although exact inference is tractable in certain classes of directed models (e.g. in tree-structured models, or models in which suitable parametric forms are chosen), for many richer models, in particular models that would seem appropriate for computer vision, exact inference is often intractable (see also section 2.1.2).

2.1.1.2 Undirected graphical models

In undirected graphical models (see also Bishop, 2006, chapter 8.3) the edges that connect the nodes of the graph are undirected and the underlying probabilistic relations between the connected random variables are symmetric. As a result, the semantics of an undirected graphical model in terms of conditional dependencies are simpler than those of a directed graphical model: Two sets of random variables A and B are conditionally independent given a third set C if all paths that connect A and B in the graph go through the set C. In particular, the "explaining away" phenomenon encountered in directed models does not exist. The *Markov blanket* of a node x_i is the set of nodes conditioned on which x_i is independent of all remaining nodes in the graph. In undirected graphical models this set contains only the immediate neighbors of a node. In

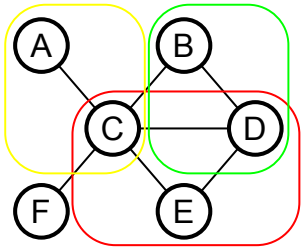


Figure 2.2: Example of an undirected graphical model. Three cliques are highlighted, two maximal cliques (yellow, red) and one non-maximal (green). The joint distribution can be written as a product of potentials defined over maximal cliques: $p(A,B,C,D,E,F;\Theta) = \frac{1}{Z(\Theta)}\Phi_{AC}(A,C;\theta_{AC})\Phi_{FC}(F,C;\theta_{FC})\Phi_{BCD}(B,C,D;\theta_{BCD})\Phi_{CDE}(C,D,E;\theta_{CDE})$.

contrast, for directed graphical models it contains the node's parents and children, but also the co-parents (the parents of children of x_i), i.e. it includes nodes that are not neighbors of the node of interest.

The graphical structure of an undirected graphical model and the way in which the corresponding distributions factorize are connected via the concept of "cliques". A clique is a fully connected subset of nodes in the graph, and the joint distribution of a Markov random field is given as a product of non-negative potential functions associated with the cliques of the graph. Without loss of generality it is often assumed that the potential functions are defined over the maximal cliques, which are all those cliques to which no further nodes can be added without the clique property being lost (see also Fig. 2.2). Denoting the set of maximal cliques by C the joint distribution over all variables can thus be written as

$$p(\mathbf{x};\Theta) = \frac{1}{Z(\Theta)} \prod_{C\in C} \Phi_C(\mathbf{x}_C;\theta_C), \tag{2.2}$$

where $\Phi_C(\mathbf{x}_C) \geqslant 0$ is the potential function associated with clique $C \in C$, and $\mathbf{x}_C$ is the subset of variables in that clique. $Z(\Theta) = \int d\mathbf{x} \prod_{C\in C} \Phi_C(\mathbf{x}_c;\theta_C)$ is the normalization constant (for the discrete case the integral is replaced by a sum over all possible states of all random variables).

A technical notion that plays an important role in the discussion of undirected graphical models and that will be used extensively throughout the thesis is that of "energy": For strictly positive potential functions the distribution defined by an undirected

graphical model is often written in terms of an *energy function* $E(\mathbf{x})$ so that

$$E(\mathbf{x}) = \sum_C \Psi_C(\mathbf{x}_C) \tag{2.3}$$

and

$$\Phi_C(\mathbf{x}_c) = \exp(-\Psi(\mathbf{x}_C)). \tag{2.4}$$

The distribution defined by the energy function $p(\mathbf{x}) = \frac{1}{Z}\exp\{-E(\mathbf{x})\}$ is also referred to as the *Boltzmann distribution*.

One appealing property of undirected graphical models is the fact that the individual potentials are not constrained to be normalized distributions or conditional distributions themselves. Compared to directed graphical models this allows for more flexibility and in many cases for more intuitive formulations, especially for problems in low-level vision. Also, due to their somewhat simpler conditional independence properties, inference in undirected graphical models with latent variables can, in certain situations, be easier than in directed models (see section 2.2.4 for an example).

The price that has to be paid for this flexibility is the normalization constant Z: Its computation is typically not tractable (for continuous $\mathbf{x}$ because the integral cannot be computed analytically; in the discrete case the total number of states might be too large for the sum to be computed). This makes it impossible to compute the *normalized* probability of a particular value of $\mathbf{x}$ exactly and therefore complicates, for instance, the assessment of the quality of a model in terms of its likelihood (estimates of the normalization constant can be obtained using techniques such as annealed importance sampling (AIS; Neal, 2001) but usually only at great computational cost). It also prevents e.g. the exact calculation of gradients with respect to the model parameters which are needed for learning. We will discuss various approaches for approximate learning in undirected graphical models in section 2.3.2 below. Furthermore, whereas directed models inherently specify a way to generate samples from the model, this is not the case for undirected models and sampling from the distribution defined by the model can be expensive.

2.1.1.3 Mixed directed and undirected models

Whether modeling goals are more easily expressed in a directed or in an undirected formulation strongly depends on the application. Although many models in the literature are either fully directed or fully undirected, in some cases it can be advantageous to combine both types of formulations. Some of the models that will be discussed

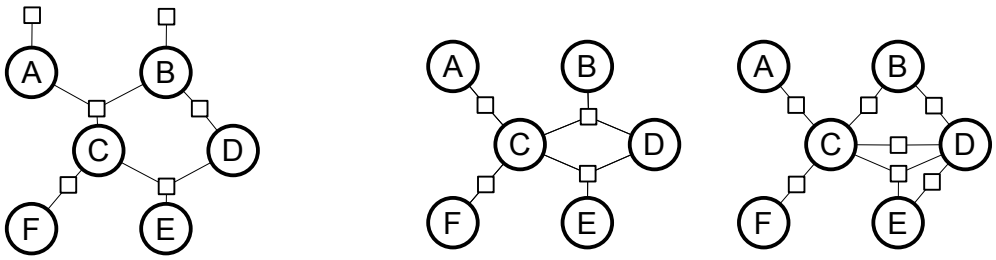


Figure 2.3: Factor graph corresponding to the directed model shown in Fig. 2.1 *(left)* and two factor graphs consistent with the undirected graphical model in Fig. 2.2 *(right)*. The factorization specified by the middle factor graph is equivalent to the factorization implied by the maximal cliques of the undirected graph in Fig. 2.2.

in this thesis are combinations of directed and undirected models, i.e. their graphical representations contain directed and undirected edges (e.g. section 3.5 as well as chapters 4 and 5). Such mixed models are also referred to as *chain graphs* and their general properties have, for instance, been studied by Lauritzen and Wermuth (1989) and Frydenberg (1990).

2.1.1.4 Factor graphs

Directed and undirected graphical models both imply a decomposition of a distribution into factors defined over subsets of the variables. In undirected models the factors are the potentials associated with the (maximal) cliques, in directed models they are the local conditional distributions (cf. equations 2.2 and 2.1 respectively). *Factor graphs* (e.g. Bishop, 2006, chapter 8.4.3) provide an alternative graphical notation that allows a more fine-grained specification of the factorization properties of a distribution by including factors directly in the graphical representation. This can be desirable especially for undirected graphical models since the potentials associated with the maximal cliques might decompose further into factors over subsets of the variables in these cliques but this will not be visible from the undirected graph. Factor graphs show factors explicitly as nodes that are drawn as squares and connected to those variables the corresponding factors are defined over. This is illustrated in Fig. 2.3, which shows two factor graphs with alternative factorizations that are both consistent with the undirected graphical model in Fig. 2.2. The factor graph for the directed model in Fig. 2.1 makes that model's Markov blanket explicit. In this thesis we will use both directed and undirected graphical models as well as factor graphs to represent models as appropriate.

2.1.2 Inference and learning

2.1.2.1 Inference

Inference in probabilistic models typically refers to computing the conditional or marginal distribution of a subset of the variables in the model, possibly given values for a second, "observed" set of variables. In particular, in a latent variable model, it refers to computing the conditional distribution over the "unobserved" (latent or hidden) variables $\mathbf{x}_H$, given the observed (or visible) variables $\mathbf{x}_V$, i.e. computing $p(\mathbf{x}_H|\mathbf{x}_V) = \frac{p(\mathbf{x}_H,\mathbf{x}_V)}{p(\mathbf{x}_V)}$. In many cases one is not interested in this distribution itself, but rather in the expectation of some function with respect to this distribution, or, in certain situations, in the value $\mathbf{x}_H^*$ that maximizes the distribution (MAP). A fundamental problem in inference is the computation of marginals over subsets of the variables. For instance, the conditional distribution $p(\mathbf{x}_H|\mathbf{x}_V)$ requires the marginal distribution $p(\mathbf{x}_V)$ as the normalization constant. Computing marginals can be very expensive: in the discrete case it requires the evaluation of the sum over $\mathbf{x}_H$ which can have an exponential number of terms, in the continuous case it can require solving intractable high-dimensional integrals. Although exact algorithms for inference exist, for instance, belief propagation for tree structured graphical models (Pearl, 1988) or the junction tree algorithm (Lauritzen and Spiegelhalter, 1988), for general graphical models this is a hard problem. In certain cases the situation can be worse for directed graphical models than it is for undirected graphical models with similar graphical structure due to the more complicated conditional independence semantics of the former and, in particular, the explaining away property. As we will see, for instance, in the models discussed below, even for seemingly very simple directed models, such as a two-layer belief network, exact inference is not tractable, whereas undirected models with very similar graphical structure admit efficient inference.

Various approximate techniques have been proposed to handle the case when exact inference is not tractable. These split broadly into two categories, deterministic and sampling based approaches. Deterministic approaches include techniques such as mean-field (MF), expectation propagation (EP), local variational approximations, or loopy belief propagation (BP) and its generalizations (see e.g. Bishop, 2006; Wainwright and Jordan, 2008 for recent overviews and comparisons of the different approaches). In mean field, an approximation to the distribution of interest, e.g. an intractable posterior distribution $p(\mathbf{x}_H|\mathbf{x}_V)$, is obtained by minimizing the Kullback-Leibler (KL) divergence (Cover and Thomas, 1991) between a tractable, approximat-

ing distribution $q(\mathbf{x}_H)$ and the distribution of interest: $\mathrm{KL}[q(\mathbf{x}_H)||p(\mathbf{x}_H|\mathbf{x}_V)]$. In the simplest case the approximating distribution can be chosen to be fully factorized, but more complex forms are possible (this is then also referred to as structured mean-field). EP also fits an approximating distribution to the distribution of interest, but in a manner different from mean-field: EP can be thought of as an iterative scheme that approximately minimizes the reverse KL-divergence $\mathrm{KL}[p(\mathbf{x}_H|\mathbf{x}_V)||q(\mathbf{x}_H)]$, giving rise to an approximation with rather different properties. Loopy BP can be seen as an application of BP (which is exact for tree-structured graphs) to graphs with cycles (where it only computes approximate marginals, and might not even converge).

The second class of techniques approximates the distribution of interest with samples (e.g. for computing the expectation of some function with respect to that distribution). In many cases, direct sampling from the distribution of interest is not possible. Simple techniques such as rejection sampling or importance sampling could be applied in these situations. Both rely on sampling from some simpler proposal distribution and then rejecting samples with a certain probability or re-weighting them so as to account for the mis-match between the proposal distribution and the distribution of interest. Both techniques are, however, often inefficient, especially in high dimensions and for distributions with mass localized in certain areas of the space. In many situations, *Markov Chain Monte Carlo* (MCMC) techniques are more efficient (see e.g. Neal, 1993 for a review). MCMC techniques construct an ergodic Markov chain whose equilibrium distribution is the distribution of interest (e.g. the model distribution or the required posterior). A Markov chain can be specified in terms of its initial distribution $p_0(\mathbf{x}_0)$ and a suitable transition kernel $T(\mathbf{x}_{t-1}, \mathbf{x}_t)$. It is simulated by first drawing $\mathbf{x}_0 \sim p_0(\cdot)$, and then repeatedly sampling $\mathbf{x}_t \sim T(\mathbf{x}_{t-1}, \cdot)$, so that at time step t the $\mathbf{x}_t$ are distributed according to

$$p_t(\mathbf{x}_t) = \int \mathrm{d}\mathbf{x}_{t-1} T(\mathbf{x}_{t-1}, \mathbf{x}_t) p_{t-1}(\mathbf{x}_{t-1}). \tag{2.5}$$

As $t \to \infty$ p_t converges towards the distribution of interest. One important question is how to choose $T(\mathbf{x}_{t-1}, \mathbf{x}_t)$. There are several approaches to constructing Markov chains, such as the Metropolis-Hastings algorithm, Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990), and Hybrid Monte Carlo (HMC; see e.g. Neal, 1993, 2011). In this thesis we will primarily use the latter two which will be discussed in more detail in section 2.3.1 below. MCMC techniques can also be used to infer the structure of a graphical model by sampling the model structure alongside the state of latent variables using, for instance, *reversible jump* MCMC (Green, 1995). One advan-

tage of sampling based approaches over deterministic approximations is that they can provide asymptotically exact results although the computation time required to achieve such results is often impractical. Deterministic approaches are often faster than sampling based approaches but the quality of the results strongly depend on the model and on the particular approximation used.

2.1.2.2 Learning

Learning typically refers to estimating the parameters of a graphical model given some data such that the data has high probability under the model. Taking a Bayesian perspective we could include prior distributions over the parameters and learning would then effectively amount to inference with respect to the posterior distribution of the parameters given the observed data. Unfortunately, for most models considered in this thesis a Bayesian approach is currently not tractable. Instead, we will consider the somewhat simpler scenario of maximum likelihood estimation. Maximum likelihood learning in general models amounts to finding values of the parameters Θ such as to maximize the likelihood (or its logarithm) of the model given the data:

$$\Theta^* = \text{argmax}_\Theta \ \log p(\mathbf{x}_V;\Theta) = \text{argmax}_\Theta \log \int \mathrm{d}\mathbf{x}_H p(\mathbf{x}_V,\mathbf{x}_H;\Theta), \tag{2.6}$$

where Θ is the set of model parameters, and $\mathbf{x}_V$ and $\mathbf{x}_H$ are observed and unobserved variables as before.

In practice, several difficulties have to be overcome: Firstly, for most latent variable models the integral over the unobserved variables cannot be computed analytically so that the expression in equation 2.6 cannot be maximized directly. Instead, schemes that alternate inference of the latent variables with maximization of the likelihood with respect to the model parameters are typically used, the most widely known of which is the expectation-maximization (EM) algorithm (Dempster et al., 1977). In its exact form EM involves two alternating steps: in the E-step the posterior distribution over the unobserved variables is obtained given a current set of values for the model parameters Θ_{old}; in the M-step the expectation of the complete data log-likelihood is maximized with respect to the model parameters:

$$\Theta_{\text{new}} \leftarrow \text{argmax}_\Theta E_{p(\mathbf{x}_H|\mathbf{x}_V;\Theta_{\text{old}})} \left[\log p(\mathbf{x}_V,\mathbf{x}_H;\Theta)\right], \tag{2.7}$$

where the expectation is taken with respect to the posterior distribution over the latent variables given the observed variables for the parameter values Θ_{old}.

Unfortunately, as discussed above, exact inference is often intractable, so that approximations have to be used. A connection between the EM algorithm and approximate inference can be obtained via the following lower bound on $\log p(\mathbf{x}_V;\Theta)$:

$$\log p(\mathbf{x}_V;\Theta) \geqslant E_{q(\mathbf{x}_H)}\left[\log p(\mathbf{x}_V,\mathbf{x}_H;\Theta)\right] + H\left[q\right] \tag{2.8}$$

where $H\left[q\right] = -E_q\left[\log q\right]$ is the (differential) entropy of q, and q is some distribution over the latent variables (e.g. Ghahramani, 1995; Saul and Jordan, 1996; Neal and Hinton, 1998; see also e.g. the discussion in Wainwright and Jordan, 2008, section 6). For learning this bound can be maximized with respect to the parameters θ and with respect to q in an alternating scheme that can be seen as a generalization of the EM algorithm. Optimization with respect to the model parameters only involves maximizing the expectation of the complete data log-likelihood as in (2.7), just that the expectation is taken with respect to q. The maximization with respect to q amounts to minimizing the KL-divergence between q and the true posterior as in mean field. If q is unconstrained then the minimum is achieved by setting q to the true posterior, i.e. to $p(\mathbf{x}_H|\mathbf{x}_V;\Theta)$. In this case the bound becomes tight and the original EM algorithm is recovered. When the true posterior is intractable a constrained form can be chosen for q (e.g. fully factorized as in standard mean field). We then no longer optimize the marginal log-probability directly, but the scheme still maximizes a lower bound. Other approximations such as EP or BP can be used as well, although in general there is no guarantee that this will result in the maximization of a lower bound of the log-probability of the data

In this thesis, we will at several points use samples from the posterior distribution to compute the expectation in (2.7), an approach also referred to as Monte Carlo EM (e.g. Wei and Tanner, 1990). A naïve application of Monte Carlo EM using MCMC to sample from the posterior can be very expensive since chains (for each data point) would need to reach their equilibrium distribution before an update of the model parameters is computed (this applies, for instance, to the models that will be discussed in chapters 4 and 5). However, as pointed out by Hinton et al. (1998) and also discussed e.g. by Teh (Teh, 2003, chapter 2.3), the generalized interpretation of the EM algorithm in terms of an optimization of the bound in (2.8) justifies a scheme that does not require chains to converge: In this view the Markov chain approximation to the posterior distribution takes the role of q. It is maintained from one iteration of EM to the next and updated by a few steps of MCMC in alternation with updates of the model parameters computed from the samples. This can be seen as a stochastic maximization of (2.8)

where updating the Markov chain approximation of the posterior using a few steps of MCMC brings it closer to equilibrium (the true posterior), thereby reducing the KL-divergence between the approximation and the true posterior and thus improving the bound with respect to q.

Even in situations in which inference with respect to the latent variables is tractable it is nevertheless often impossible to maximize the likelihood exactly. This is, for instance the case for undirected graphical models for which the intractability of the normalization constant poses a serious problem. The normalization constant is a function of the model parameters so it and its gradients are required to maximize the likelihood. A range of approaches have been developed to deal with this problem. Some approaches propose alternative inductive criteria for parameter estimation and thus sidestep the problem of computing the normalization constant and its derivatives, while others approximate the intractable terms, e.g. using samples. We will discuss some of these approaches in more detail in section 2.3 below.

Finally, other problems might be encountered. For instance, if the structure of a model has to be learned in addition to its parameters, this can require a very expensive search over a larger number of alternative model structures which is often not feasible in practice (e.g. Friedman, 1997). In other cases it might not be possible to jointly learn all the parameters simultaneously since the resulting optimization problem is highly non-convex so that learning is unlikely to find a sufficiently good local optimum. A range of learning strategies such as greedy learning or strategies that gradually increase the complexity of the model class or of the learning problem have been developed which can mitigate some of these problems in certain situations (e.g. Williams and Titsias, 2004; Hinton et al., 2006b; Kumar et al., 2010; Bengio et al., 2009).

2.2 Probabilistic models for low- and mid-level vision

In this section we will review several classes of generative models that have been used to model low- and mid-level structure in vision. The focus will be on the technical aspects of those models that form the context of the work presented in the subsequent chapters. The models that we will review here have been developed in different lines of the literature, mainly in the computer vision and machine learning literature, but also, for instance, in neuroscience. They have been motivated by a range of different prob-

lems, including image processing tasks such as denoising, computer vision problems such as object recognition, or by the goal to characterize the statistical properties of natural images for the purpose of developing efficient image codes and understanding properties of sensory processing in biological systems. Accordingly, different models focus on different aspects of the modeling problem, and employ different model formulations.

From a technical perspective, the formulations vary in several ways including whether they are directed, undirected, or mixed formulations; whether they are defined over images of fixed size or can be applied to images of arbitrary size; whether they are formulated in terms of latent variables (as we will see below some models can be interpreted either way) and how these latent variables are used; and whether they are shallow or hierarchical[1]. As we will see, the ease (or difficulty) with which learning and inference can be performed depends strongly on these properties. On a more conceptual level the models differ with respect to the implicit and explicit assumptions they make about the image formation process, and thus with respect to the type of image structure that they can model well: Most models that we will discuss are generic learning architectures but others are highly structured formulations and make very specific assumptions about the nature of structure in images and how this structure should be represented. Furthermore, all models define a probability distribution over the space of images and thus characterize the statistical regularities images. But while some do so in terms of a "black-box" density model, others additionally provide a latent representation that captures important aspects of the structure in an image in a manner that is more interpretable than the raw image intensities themselves, and that can thus be useful for a range of computer vision tasks. (In fact, in some cases the main purpose of learning a generative model of some dataset is to learn a set of features that can subsequently be used discriminatively, e.g. as input to classifiers.)

Overall the discussion will be somewhat biased towards undirected models since these are the main building blocks of the work described in the following chapters. We will first discuss homogeneous MRFs (cf. section 2.2.1.1), which have a long tradition as image priors in computer vision and image processing, as well as their directed counterparts, causal random fields (section 2.2.1.2). Section 2.2.2 then covers "sparse coding" and related models, a class of directed latent variable models which have their

[1] In the context of this thesis the term "hierarchical" is used to refer to models with multiple layers of latent variables, often with an architecture that is replicated across layers of the hierarchy. This use is somewhat different from the use in the context of "hierarchical Bayesian models" where it typically refers to a cascade of prior distributions over model parameters.

origin in the study of natural image statistics. Unlike the random field models these models have been largely limited to small image patches. Products-of-Experts (PoEs; section 2.2.3) for image modeling have been motivated in a manner similar to the sparse coding models but due to their undirected nature have favorable inference properties compared to the former. Restricted Boltzmann Machines (RBMs; section 2.2.4) are a special type of PoE model and have recently gained popularity due to the fact that they can be used as building blocks for efficient learning of hierarchical architectures. Such hierarchical architectures, especially work from the "deep learning" community, which emphasizes the advantage of hierarchical, distributed representations, will be discussed in section 2.2.5. Some examples of more structured representations will be briefly discussed in section 2.2.6.

In the remainder of this section we will denote the image using $\mathbf{x}$, where x_{ij} indicates the value of the pixel at position (i, j) (in many cases we will use only a single index i to keep the notation uncluttered). Different models assume x_{ij} to be either discrete or continuous, and this should become clear from the context.

2.2.1 Models of dense fields

Models of dense fields are probably the most popular models in low-level vision, and they have been used extensively as priors for image processing tasks such as denoising, inpainting, or optical flow estimation, but also, for instance, for texture modeling. The distinctions between the models discussed in this section and those discussed later in the chapter are in some cases a bit blurred, but one distinguishing feature of the models discussed here is that they are all stationary and can be applied to images of arbitrary size. Furthermore, they are typically formulated without latent variables. We will discuss the more common undirected formulation first (section 2.2.1.1); directed formulations will then be discussed in section 2.2.1.2.

2.2.1.1 Homogeneous MRFs

The general use of the term "MRF" is simply as an alternative to "undirected model". In many cases, however, in particular in low-level vision, it is understood to refer to an undirected model of a particular form: The nodes of the model form a two-dimensional lattice that reflects the spatial organization of the data to be modeled (e.g. the pixels of an image); the cliques are typically defined on relatively small subsets of nearby nodes in this lattice; and the clique-structure is replicated across the lattice so that

the constraints imposed by the MRF are invariant to the spatial location (ignoring the difficulties arising at the image boundaries which often necessitate special treatment). This reflects the idea that many properties of natural images are independent of the image position, and it allows modeling images of arbitrary size with a model that has a relatively small number of parameters. MRFs with such a replicated clique structure are referred to as *homogeneous* or *stationary* and have a long history in computer vision (e.g. Geman and Geman, 1984; Marroquin et al., 1987; Szeliski, 1990; see also Roth, 2007, sections 2.2.2 and 2.2.3 for a recent review).

The general form of a homogeneous MRF is given by

$$p(\mathbf{x}) \propto \prod_{(i,j)} \prod_{k=1}^{K} \Phi_k(\mathbf{x}_{N_k(i,j)}; \theta_k) \tag{2.9}$$

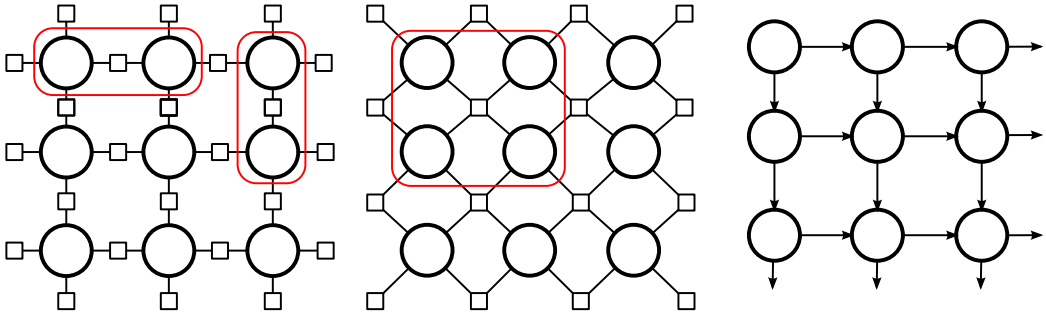
where (i, j) indexes the position in the two-dimensional lattice, k indexes the different clique types, and $N_k(i, j)$ define the neighborhood structure of the MRF: K cliques are centered at each node (i, j), and the neighborhoods are defined relative to that node. (In many cases there is only a single type of clique.) Note that in this formulation the potential functions and their parameters do not depend on the position in the lattice and that the neighborhood structure remains invariant.

Homogeneous MRFs can differ with respect to their neighborhood structure and with respect to the potential functions used. A large body of literature deals with a particularly simple type of neighborhood, where the cliques are formed by pairs of typically nearby nodes. For instance, a simple four-connected MRF in which each node is connected to its immediate horizontal and vertical neighbors (cf. Fig. 2.4a) would take the form

$$p(\mathbf{x}) \propto \prod_{(i,j)} \Phi_H(x_{i,j}, x_{i,j+1}) \Phi_V(x_{i,j}, x_{i+1,j}) \tag{2.10}$$

where we have assumed that the potential functions are symmetrical (e.g. $\Phi_H(x_{i,j}, x_{i,j+1}) = \Phi_H(x_{i,j+1}, x_{i,j})$).

The widely known Ising model for binary random variables (i.e. $x_i \in \{-1, 1\}$), and its generalization to categorical random variables with more than two states, the Potts model, are two examples of pairwise MRFs for discrete random variables. They are used, for instance, as priors over label images for segmentation tasks, in which case Φ is chosen such as to penalize label differences between nearby sites (in the case of the Ising model, for instance, $\Phi(x_i, x_j) = \exp(\beta x_i x_j)$ with $\beta > 0$). Common forms of pairwise potentials for continuous variables include the squared exponential



(a) (b) (c)

Figure 2.4: (a) Factor graph of a homogeneous MRF with pairwise cliques; (b) factor graph of a homogeneous MRF with high-order clique; (c) causal MRF. ((a,b) adapted from Roth, 2007)

$\Phi(x_i, x_j) = \exp(-\alpha(x_i - x_j)^2)$ for some $\alpha > 0$, giving rise to Gaussian MRFs (e.g. Woods, 1972), or robust, "discontinuity-preserving" potentials frequently used in simple image priors (see e.g. Geman and McClure, 1985, for an example).

Pairwise MRFs are widespread but they can be too limited for certain applications (cf. e.g. Morris et al., 1996; Tjelmeland and Besag, 1998; Geman and Reynolds, 1992; see also the discussion in Roth, 2007, section 2.2.2, and in Roth and Black, 2009). Most of the work in this thesis will be concerned with *high-order* MRFs. Here, the potentials are defined over larger numbers of pixels, and for most purposes one can think of the neighborhood $N_k(i, j)$ that defines a clique as a small image patch centered at pixel (i, j). Using a single index to denote each node in the lattice we will below also use the notation $\mathbf{x}_{(i)}$ to indicate the image patch centered at pixel (node) i, so that equation (2.9) becomes

$$p(\mathbf{x}) \propto \prod_i \prod_k \Phi_k(\mathbf{x}_{(i)}). \tag{2.11}$$

As for pairwise MRFs different types of high-order potential functions are being used in the literature, for continuous valued (e.g. Zhu and Mumford, 1997; Zhu et al., 1998; Roth and Black, 2005) as well as for discrete random variables (e.g. Tjelmeland and Besag, 1998; Rother et al., 2009). A common form for continuous data is a potential that is formulated as a scalar function operating on a one-dimensional projection of the data, typically the response of a filter with limited support. Several examples of this type of clique potential will be discussed in more detail in chapter 3 and in section 2.2.3 below. Fig. 2.4b shows an exemplary factor graph where each clique is defined over 2×2 "patches".

As for most undirected graphical models evaluation of the likelihood and exact maximum likelihood learning are impossible due to the intractability of the normalization constant, even in the fully observed case. In many cases, MRF parameters are therefore not learned but rather chosen heuristically; if they are learned, approximate techniques usually have to be used. The FoE (Roth and Black, 2005), for instance, which we discuss in chapter 3, is trained using contrastive divergence (see section 2.3.2). The nature of the representation learned by a MRF can be hard to interpret, especially for high-order MRFs (e.g. section 3.3.1), and generating samples from an MRF model typically relies on MCMC techniques (such as Gibbs sampling, cf. section 2.3.1 below), and can be very time consuming and frequently suffers from poor mixing of the chains.

2.2.1.2 Causal random fields

Although undirected formulations are by far predominant, directed formulations, causal random fields, also exist. In this case, an ordering is introduced for the nodes in the lattice so that the joint distribution can be factored as in equation (2.1). If the conditional probability of a pixel given its causal neighborhood is independent of the image position again a stationary model is obtained: $p(\mathbf{x}) = \prod_i p(x_i|\mathbf{x}_{\text{causal}(i)};\theta)$, where $\mathbf{x}_{\text{causal}(i)}$ denotes the variables in the causal neighborhood of node i. An example of such a causal neighborhood is shown in Fig. 2.4c. An early mention of causal random fields for pattern recognition problems can be found in Abend et al. (1965) where they are referred to as *Markov mesh* models. More recently they have been applied to texture or image modeling problems e.g. by Popat and Picard (1993), Gray et al. (1994), or Domke et al. (2008). Compared to homogeneous MRFs these models have the advantage that, in the fully observed case, learning is exact and fast, and that the likelihood can be evaluated exactly without having to resort to expensive sampling based techniques for estimating the normalization constant. Also, samples from the model are easily generated using ancestral sampling. Nevertheless, causal RFs are far less widely used than their undirected counterparts. It is interesting to note that many of the powerful non-parametric models in which new images are composed by stitching together patches from a ground truth image (e.g. Efros and Leung, 1999; see also discussion in section 3.3.2) can be interpreted in this framework.

2.2.2 Sparse coding and related directed models

The models discussed in section 2.2.1 are distinctive in that they are translation invariant and can thus be applied to images of arbitrary size. Furthermore, they do not usually involve latent variables. In this section we will discuss a class of models that employ latent variables in order to introduce dependencies between image pixels. Many of the models described in this section have their origins in the computational neuroscience literature, and are motivated by the idea that important properties of the human visual system can be understood in terms of the statistical properties of natural images and the system's need to transform the raw visual input into a more efficient representation for further processing (see e.g. Barlow, 1989). This is achieved by describing the data with a generative model that explains structure in images in terms of latent "causes", and perception then corresponds to inferring the relevant causes given the evidence provided by an image. A common assumption is that an image is generated by selecting a small number of *independent* causes out of a possibly very large dictionary, which then interact to form the image. An important hope is that given these assumptions, unsupervised learning will allow the model to recover the dictionary of latent causes underlying a given dataset. The idea that only a small number of causes contributes to any given image is also referred to as *sparsity*, and the large size of the dictionary (the number of causes can be much larger than the number of input dimensions) as "over-completeness". One of the best known models in this group is the sparse coding model proposed by Olshausen and Field (1997), but there are many notable variations and extensions of this work (e.g. Bell and Sejnowski, 1997; Lewicki and Olshausen, 1999; Hyvärinen and Hoyer, 2000; Hyvärinen et al., 2001; Karklin and Lewicki, 2009; Sinz et al., 2010; Puertas et al., 2010), which will be discussed in more detail in relation to the masked RBM in chapter 4 (section 4.4.1).

A common implementation of the ideas of independence, sparsity, and overcompleteness is in terms of a sparse, independent prior distribution over the latent variables and a linear-Gaussian conditional distribution over the image. For instance, the model proposed by Olshausen and Field (1997) is given as follows:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} \tag{2.12}$$

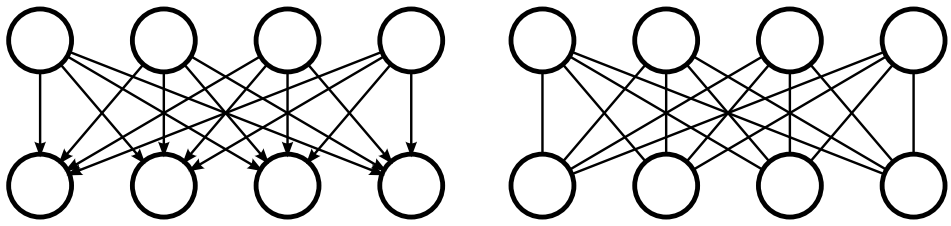
$$p(\mathbf{x}|\mathbf{z}) = N(\mathbf{x}|W\mathbf{z}+\boldsymbol{\mu},\sigma^2\boldsymbol{I}) \tag{2.13}$$

$$p(\mathbf{z}) = \prod_j p(z_j), \tag{2.14}$$

where $\mathbf{z}$ is the vector of latent variables and W is a matrix containing basis functions,

in its columns. $p(z_j)$ is some univariate sparse distribution, e.g. a Laplace or Cauchy distribution. The corresponding graphical model is shown in Figure 2.5(a). This formulation allows for a larger number of basis functions than the dimensionality of the image (over-completeness), and encoding (i.e. inferring the posterior over the latent variables $\mathbf{z}$ given an image $\mathbf{x}$) is then a nonlinear process. A Gaussian prior would give rise to the simpler probabilistic principal component analysis model (PPCA; Tipping and Bishop, 1999) which is computationally more convenient but only able to model (linear) second-order correlations in the data. Independent Component Analysis (ICA; Bell and Sejnowski, 1997) can be considered as a special case in which the variance of the observation noise goes to zero and the number of basis functions is the same as the dimensionality of the image, so that W is square and invertible and inference amounts to a deterministic transformation.

These causal models have an intuitive generative process. When trained on natural image patches the models have a tendency to learn Gabor-like basis functions that resemble the receptive field properties of simple cells in the visual cortex (e.g. Olshausen and Field, 1997). Also, the sparse latent representations $\mathbf{z}|\mathbf{x}$ can perform well in recognition tasks (e.g. Raina et al., 2007). Nevertheless, important assumptions such as the strict independence of the latent variables (eq. 2.14) and the linear combination of the bases (eq. 2.13) are overly simplistic, limiting the models' ability to provide a good statistical description of images and to recover "meaningful" causes (see also discussion in section 4.4.1). A practical disadvantage is further that for most interesting forms of the prior and the likelihood (in particular for most sparse priors) exact inference is intractable, except for ICA, i.e. in the case when the number of latent and observed variables is the same (e.g. Bell and Sejnowski, 1997). It is readily seen from the graphical model that the latent variables will be conditionally *dependent*, and an analytic form for the posterior does not exist. This also causes problems during learning, and approximate schemes have to be used (a common approach is to use the mode of the posterior instead of the full distribution in an EM-like scheme, an approach that necessitates additional constraints on the parameters to work; Berkes et al., 2007 employ a variational scheme; Seeger, 2008 applies EP). The difficulties associated with inference and learning are one reason why many models in this class have so far remained limited to single-layer (i.e. non-hierarchical) representations and to small image patches (non-stationary formulations), although some hierarchical formulations have been considered (see e.g. Karklin and Lewicki, 2003; Sinz et al., 2010) and several recent works investigate translation invariant, convolutional formulations



(a) (b)

Figure 2.5: Graphical models of (a) sparse coding model, and (b) RBM

of models in this class (e.g. Kavukcuoglu et al., 2010; Zeiler et al., 2010; see also Smith and Lewicki, 2005).

2.2.3 Products of Experts

A class of models whose application to images has been motivated in a manner similar to the "sparse coding" models discussed in the previous section is formed by the "product of experts" (PoE) models. The term PoE was introduced by Hinton (2002) and it refers to *undirected* models that can be written as a product of component models (the experts)

$$p(\mathbf{x};\theta_1 \dots \theta_K) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K} \Phi_k(\mathbf{x};\theta_k), \tag{2.15}$$

where $Z(\Theta) = \int \mathrm{d}\mathbf{x} \prod_k \Phi_k(\mathbf{x}|\theta_k)$ is the normalization constant and θ_k are the parameters of the individual experts (Θ is the set of parameters of all K experts). The individual experts $\Phi_k(\mathbf{x}|\theta_k)$ typically take the form of unnormalized probability distributions or densities. Unlike in a mixture where each data point is generated from only one of the mixture components, in a PoE the experts interact and jointly shape the distribution defined by their product which can therefore be a lot tighter than the distribution of each individual expert. Even a product of individually simple experts can jointly define a complex, multi-modal distribution (an example of this is shown in Fig. 3.12, page 76). Individual experts do not have to define a distribution over the full data space but can act only in some directions or on a subset of the dimensions of the space.

The notion of a PoE is very general and many probabilistic models in the literature can be interpreted in that way. One example is, for instance, minor component analysis (MCA) proposed by Williams and Agakov (2002) in which the individual experts are one-dimensional Gaussians (the distribution defined by their product is thus also Gaussian) and which will be discussed in more detail in chapter 3.2.2. There is a lot of

freedom with respect to the choice of the experts, and, in particular, the experts can be defined in terms of latent variables (see e.g. Hinton, 2002, for several examples).

Teh et al. (2003) and Osindero et al. (2006) apply PoE models to natural images and discuss their connection to causal models such as sparse coding and ICA. When modeling natural images, the experts are often defined in terms of one-dimensional projections of the image. For instance, in Teh et al. (2003) they take the form

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K} \Phi(\mathbf{w}_k^T \mathbf{x}; \boldsymbol{\alpha}_k) \tag{2.16}$$

where the image $\mathbf{x}$ is as a vector of length N, i.e. $\mathbf{x} \in \mathbb{R}^N$, and each expert is defined in terms of $\mathbf{w}_k^T \mathbf{x}$ where $\mathbf{w}_k$ defines the subspace of expert $k = 1 \dots K$ (K is the number of experts). $\Phi(y; \boldsymbol{\alpha}_k)$ is an nonlinear expert function with parameters $\boldsymbol{\alpha}_k$ (typically an unnormalized 1D density function) and Θ is the set of parameters of the model (basis vectors $\mathbf{w}_k$s and expert parameters $\boldsymbol{\alpha}_k$s). A good way to think about the workings of this model is in terms of constraints: Each of the experts effectively constrains the distribution in one direction in image space, the direction being defined by $\mathbf{w}_k$. The nature of the constraint that is imposed depends on the "expert" function $\Phi(y; \boldsymbol{\alpha}_k)$. Osindero et al. (2006) consider an extension of this model in which each expert is defined in terms of a linear combination of squared projections $(\mathbf{w}_k^T \mathbf{x})^2$.

Choosing the expert function $\Phi(y; \boldsymbol{\alpha}_k)$ to be an unnormalized, heavy-tailed density function such as the Student-t density function in Teh et al. (2003), the model bears much resemblance to the causal models discussed in the preceding section (2.2.2): In this case each expert can also be formulated in terms of a Gamma-distributed latent variable (see Teh et al., 2003; Osindero et al., 2006, for details). The corresponding graphical model is shown in Fig. 2.5(b) which highlights a general property of PoE models with latent variables: conditional on the observed data (here the image $\mathbf{x}$) the latent variables of all experts are independent. Thus, whereas in directed models the latent causes are marginally independent but conditionally dependent given an image, the opposite is true for PoE models: the latent units are conditionally independent but marginally dependent. In the causal models discussed in the previous section sampling from the model is cheap but inference expensive. In PoEs, inference is cheap – but generating new samples from the model can be expensive. Learning of PoE models involves solving the same problems as for other undirected graphical models and will be discussed in more detail below. Square ICA which we considered above as a special case of sparse coding models can also be seen as a special case of a PoE model.

It has the particularly appealing property that the normalization constant can be computed analytically. When trained on natural images sparse PoE models as in Teh et al. (2003) and Osindero et al. (2006) learn filters that resemble Gabor wavelets and are thus very similar to the basis functions learned by causal models discussed in the previous section. Each of these filters imposes a smoothness constraint on the data that is usually approximately satisfied, but occasionally violated (if the filter is co-located with an edge in the image; see Hinton and Teh, 2001). The latent activation of an overcomplete Student-t PoE is also sparse, but typically less so than for a similar directed model due to the difference in inference, and it is also usually more stable with respect to small changes in the image $\mathbf{x}$ (Osindero et al., 2006). An interesting representational difference between PoEs and directed models can be seen when considering the simple Gaussian case: the causal PPCA model (see previous section) identifies the subspace of *highest* variation in the data (the principal components); in contrast, in MCA, the Gaussian experts identify directions of exceptionally *small* variation (and constrain the Gaussian distribution in those directions).

The homogeneous MRFs discussed in section 2.2.1 are special cases of PoE models: in that case each potential is an expert that imposes a constraint on the joint configuration of nearby pixels. Each expert is defined only on a subset of the dimensions of the data space (the nodes in its clique) and the experts (potentials) at different image locations share parameters. Fully connected PoE models (with potentials defined over all observed variables, e.g. the full image) are typically only applied to relatively small image patches and not to larger images.

2.2.4 Restricted Boltzmann Machines

2.2.4.1 Restricted Boltzmann Machine for binary data

A class of undirected models that is of particular importance for this thesis is the Restricted Boltzmann Machine (RBM; Smolensky, 1986; Freund and Haussler, 1994), which is a special case of the Boltzmann Machine (BM; Ackley et al., 1985). The BM is an undirected graphical model with binary random variables $s_i \in \{0,1\}$ whose probability distribution is given by an energy function

$$E(\mathbf{s}) = -\sum_{i,j} w_{ij} s_i s_j - \sum_i \theta_i s_i, \tag{2.17}$$

where w_{ij} is the connection strength between unit i and unit j and θ_i is the unit's threshold, or bias.

In the *restricted* BM, which Smolensky also referred to as the "Harmonium model", the units are partitioned into two sets, the "visible" units, which are usually denoted by v_i, and the "hidden" units, denoted by h_j. Connectivity is restricted such that visible units are connected only to hidden units (and vice versa; the model is undirected), but there are no connections between visible units or between hidden units. The graphical model thus forms a bi-partite graph as shown in Fig. 2.5 and the energy function of this formulation is given by

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i,j} w_{ij} v_i h_j - \sum_i v_i b_i - \sum_j h_j c_j \tag{2.18}$$

$$= -\mathbf{v}^T W \mathbf{h} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h} \tag{2.19}$$

where the hidden and visible biases are now denoted by $\mathbf{b}$ and $\mathbf{c}$ respectively. The corresponding joint distribution takes the usual form $p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\{-E(\mathbf{v},\mathbf{h})\}$ where $Z = \sum_{\mathbf{v},\mathbf{h}} \exp\{-E(\mathbf{v},\mathbf{h})\}$ is the normalization constant, which cannot be computed when the number of hidden units and the number of visible units are both large since it involves a sum over $2^{\min(N,M)}$ configurations of the visible or hidden units (N: number of visible units, M: number of hidden units). In most cases one is interested in fitting the *marginal distribution over the visibles* to some data set. This marginal distribution is given by

$$p(\mathbf{v}) = \frac{1}{Z}\sum_h \exp\{-E(\mathbf{v},\mathbf{h})\}. \tag{2.20}$$

The bipartite structure and restricted connectivity of the RBM makes inference easy. As can be seen directly from the graphical model in Fig. 2.5 units in one set are conditionally independent given the all the units in the second set:

$$p(\mathbf{v}|\mathbf{h}) = \prod_i p(v_i|\mathbf{h}) = \prod_i \frac{1}{1+\exp(-W_{i\cdot}\mathbf{h} - b_i)} \tag{2.21}$$

$$p(\mathbf{h}|\mathbf{v}) = \prod_j p(h_j|\mathbf{v}) = \prod_j \frac{1}{1+\exp(-W_{\cdot j}^T\mathbf{v} - c_j)}, \tag{2.22}$$

where $W_{i\cdot}$ and $W_{\cdot j}$ denote the ith row and jth column of W respectively.

The binary RBM has recently received a lot of attention, mainly due to the fact that it admits for fast inference and approximate learning and because it is the building block for greedy learning of "deep" architectures (Hinton et al., 2006b; see also section 2.2.5 below). Furthermore, through the hidden units the RBM is able to model high-order dependencies between the visible units and it can approximate arbitrary binary distributions (although at the cost of a potentially exponentially large number of hidden units; Freund and Haussler, 1994; Le Roux and Bengio, 2008).

There are two ways of thinking about the marginal distribution over the visibles defined by the RBM. Firstly, it can be thought of as a mixture of multivariate Bernoulli distributions where the number of mixture components is exponential in the number of hidden variables (cf. eq. 2.20). Alternatively, it can be considered as a product of experts. Each hidden unit gives rise to one expert and each of these experts is a mixture of a uniform distribution (the hidden unit is off) and a multi-variate Bernoulli distribution (when the hidden unit is on). This view also allows summing out one set of variables analytically:

$$p(\mathbf{v},\mathbf{h}) \propto \exp(\mathbf{v}^T W\mathbf{h} + \mathbf{b}^T\mathbf{v} + \mathbf{c}^T\mathbf{h}) \tag{2.23}$$

$$= \exp(\mathbf{b}^T\mathbf{v}) \prod_j \exp(\mathbf{v}^T W_{\cdot j} h_j + c_j h_j) \tag{2.24}$$

$$p(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h})) \tag{2.25}$$

$$= \frac{1}{Z}\exp(\mathbf{b}^T\mathbf{v}) \prod_j \sum_{h_j} \exp(\mathbf{v}^T W_{\cdot j} h_j + c_j h_j) \tag{2.26}$$

$$= \frac{1}{Z}\exp(\mathbf{b}^T\mathbf{v}) \prod_j \left[1 + \exp(\mathbf{v}^T W_{\cdot j} + c_j)\right] \tag{2.27}$$

(see also Freund and Haussler, 1994). An equivalent factorization exists with respect to the visible units:

$$p(\mathbf{v},\mathbf{h}) \propto \exp(\mathbf{c}^T\mathbf{h}) \prod_i \exp(v_i W_{i\cdot}\mathbf{h} + b_i v_i). \tag{2.28}$$

The RBM can thus be seen either as an inhomogeneous pairwise MRF (with latent variables), or as a fully connected MRF with high-order potentials involving all visibles (without latent variables). The form in equation (2.27) is of importance since it allows the efficient computation of the *unnormalized* probability of a data vector $\mathbf{v}$ under the model (although for computing the normalized probability we are still lacking the normalization constant), a property we will make use of in chapters 4 and 5.

2.2.4.2 RBMs for non-binary data

The RBM, in the basic form presented above defines a distribution over binary variables but it can be extended to other types of random variables. Welling et al. (2004), for instance, discuss their extension to exponential family distributions. One form that has been used repeatedly in the literature for modeling continuous data is a RBM with Gaussian visible and binary hidden units (Freund and Haussler, 1994; Lee et al., 2009):

$$E(\mathbf{v},\mathbf{h}) = \frac{1}{2\sigma^2}\mathbf{v}^T\mathbf{v} - \frac{1}{\sigma^2}\left(\mathbf{b}^T\mathbf{v} + \mathbf{c}^T\mathbf{h} + \mathbf{v}^T W\mathbf{h}\right). \tag{2.29}$$

In this model the conditional distributions over the visible units are Gaussian distributions with mean determined by the state of the hidden units but with fixed variance. Overall, this RBM therefore defines a mixture of isotropic Gaussians. The fact that the variance of the mixture components is fixed is a serious limitation. Recently, several alternative continuous valued RBMs have been proposed (e.g. Ranzato and Hinton, 2010; Nair and Hinton, 2010; Le Roux et al., 2011; Courville et al., 2010; see also Welling et al., 2004). In chapter 4 we will discuss and make use of one particular form, a RBM where the conditional distributions over the visible units are Beta distributions (Le Roux et al., 2011). In this RBM the mean *and* variance of the conditional distribution over the visibles *both* depend on the state of the hidden units. Another form of RBM for discrete visible units is the categorical or "softmax" RBM. Categorical visible units are often used to model labels (e.g. Hinton et al., 2006b). In the general case the energy of a RBM with categorical visible units with K states is given by

$$E(\mathbf{v},\mathbf{h}) = -\sum_k \mathbf{v}_k^T W_k \mathbf{h} - \sum_k \mathbf{b}_k^T \mathbf{v}_k - \mathbf{c}^T \mathbf{h}, \tag{2.30}$$

where the i-th categorical visible unit is written in terms of K binary units v_{ki} with the additional constraint that for each i these k binary units are mutually exclusive: $v_{ki} = 1 \;\Rightarrow\; v_{k'i} = 0 \;\forall k' \neq k$. In chapters 4 and 5 we will discuss a special form of this categorical RBM.

2.2.4.3 RBMs and homogeneous MRFs

For computational reasons, and because the number of parameters grows quite rapidly most applications of RBMs to images or image-like data attempt to model relatively small patches. However, some formulations restrict the connectivity between hidden units and visible units in such a way that each hidden unit is connected only to a relatively small set of adjacent visible units (i.e. each hidden unit has a limited "receptive field"), and also introduce weight sharing between hidden units that are connected to equivalent sets of visible units but at different positions in the image (Desjardins and Bengio, 2008; Lee et al., 2009; Ranzato et al., 2010b; Norouzi et al., 2009). In the simplest form (e.g. in Desjardins and Bengio, 2008), this gives rise to what is effectively a homogeneous MRF in which the clique potentials are unnormalized RBMs defined over the local image patch. Lee et al. (2009); Norouzi et al. (2009) use a convolutional formulation in the context of hierarchical models in which convolutional layers are alternated with probabilistic sub-sampling (max-pooling) layers similar to convolutional

neural networks, thereby achieving spatial feature pooling. Ranzato et al. (2010b) evaluate the impact of different weight sharing schemes and find that fully convolutional schemes might be overly restrictive and propose a "tiled-convolutional" scheme instead, in which units with the same sets of weights are replicated only every N pixels.

2.2.5 Hierarchical models and deep learning

Tree structured hierarchical models, for which efficient inference is possible, have a long tradition in computer vision, in particular for modeling the part based structure of objects (see discussion in section 2.2.6 below), but also for modeling low level structure. Such models can be used as alternatives to MRF models. Examples include the tree-structured formulations proposed by e.g. Luettgen and Willsky (1995) and Bouman and Shapiro (1994). The hidden Markov tree model described by Bouman and Shapiro, for instance, introduces correlations between neighboring pixels by conditioning nearby variables on shared parents at a coarser scale. The tree structure keeps inference efficient but also leads to artifacts since the dependency structure defined by the model is no longer homogeneous and not necessarily well aligned with the structure in a given image. This problem is addressed by models that allow for an *adaptive* tree structure, e.g. Williams and Adams (1999), but at the expense of making inference considerably harder.

In tree-structured models each unit (latent variable) only has a single parent. In *dense* hierarchical models this constraint does not exist and units are connected to many parents. Such models have received less attention, not least because learning and inference is hard. One early piece of work that extends a dense causal model hierarchically is the Helmholtz machine (Dayan et al., 1995), whose generative model is a two-layer sigmoidal belief network, i.e. a two-layer model with binary units and dense connectivity (cf. Fig. 2.6). Since all latent variables are conditionally dependent given an image, exact inference in such a model is very difficult. The model therefore has an associated "recognition network" (a neural network with sigmoidal units), which allows the efficient approximation of the posterior distribution over the hidden variables. This can be thought of as a variational approach that uses an approximate posterior distribution that factorizes in each layer. Training the model involves learning two sets of weights, the "generative" weights (i.e. the parameters of the causal generative model, the sigmoidal belief network), as well as the "recognition weights" (the variational parameters of the approximate posterior distribution). Simultaneous learning of

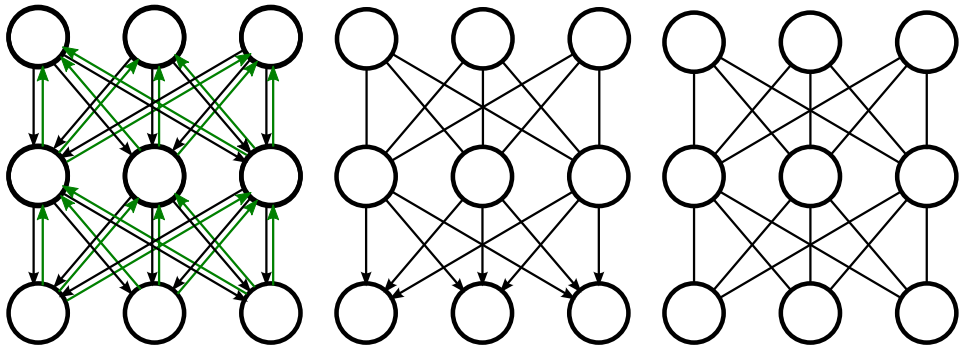


Figure 2.6: (a) Helmholtz machine, (b) DBN, (c) DBM. In (a), the black arrows show the generative connections (the generative model is simply a multi-layer sigmoidal belief-network); the green arrows depict the associated recognition network. Note that in (b) the connections in the top layer are undirected, i.e. this model is a chain graph.

both sets of weights will lead to a recognition model that is a good approximation of the true posterior – and at the same time will lead to a generative model whose true posterior distributions are close to the factorial approximation provided by the recognition model. The learning algorithm for the Helmholtz Machine is also referred to as "wake-sleep algorithm" (Hinton et al., 1995). The graphical model corresponding to the Helmholtz Machine is shown in Fig. 2.6(a), with the generative part being drawn in black.

Following Hinton et al. (2006b) models related to the Helmholtz machine have re-gained considerable popularity, leading to the development of a range of "deep" (hierarchical) generative models with dense connectivity and distributed latent representations. Hinton et al. (2006b) propose using the RBM as an elementary building block to greedily learn multi-layer models, one layer at a time: Once a first RBM has been trained on the data, the latent representation of the data is inferred, and this set of latent binary vectors is then again modeled using a binary RBM. As shown in Hinton et al. (2006b) adding a second layer is guaranteed to improve the likelihood of the model or at least leave it unchanged if the weights of the second layer RBM are initialized in a suitable manner. Greedy pre-training can be followed by a joint training phase in which all model components are trained together, to optimize generative or discriminative performance. Hierarchical models that can be trained in this manner include *Deep Belief Networks* (DBN; Hinton et al., 2006b), which are lay-

ered sigmoidal Belief Networks with undirected connections between the two sets of units in the top two layers, and *Deep Boltzmann Machines* (DBM; Salakhutdinov and Hinton, 2009), their fully undirected equivalent (see also Fig. 2.6b,c). Both types of models have recently been used as generative models of various types of data (including images, e.g. Salakhutdinov and Hinton, 2009; Ranzato et al., 2011), but also to learn feature hierarchies to be used discriminatively, e.g. for recognition. The hope associated with increasing the depth of the hierarchy is that this improves the quality of the generative model, and leads to more invariant representations that reflect more abstract properties of the data in the higher layers and thus lead, for instance, to improved performance in recognition tasks. Such an improved discriminative performance has indeed been found in at least some applications (e.g. Dahl et al., 2010) although the directly measured invariance appears to be only moderate (e.g. Goodfellow et al., 2009). Quantitative analyses in terms of the log-likelihood are made difficult by the fact that these models are at least partially undirected and by the presence of latent variables. Interestingly, several analyses that have been performed suggest that increasing the depth of the hierarchy leads, at best, only to small improvements (e.g. Salakhutdinov and Murray, 2008; Salakhutdinov and Hinton, 2009; Murray and Salakhutdinov, 2009; Theis et al., 2010).

2.2.6 Structured representations

Most of the models discussed in the previous sections can be considered as generic learning architectures and make relatively few and general assumptions about the nature of structure in images. Instead of imposing a particular representation they attempt to learn a suitable representation from the statistical regularities in the data.

Other models make stronger assumptions and are formulated in terms of directly interpretable entities. This approach is more prevalent in the context of models of higher-level structure (e.g. modeling the part-based, hierarchical structure of objects) or in more restricted domains (see, for instance, the discussion of layered image models in section 4.4.2 of chapter 4). Nevertheless, some applications in low- and mid-level vision also exist. As an example, we consider here the work by Guo et al. (2003) in which an image is defined in terms of a "texton map" (or multiple such maps). These texton maps consist of random numbers of textons, which are defined as deformable templates, and each texton has a set of attributes defining, for instance, its position, orientation, and scale, or its appearance. Within each map an *inhomogeneous* MRF

imposes mutual constraints on nearby textons with respect to the relative values of these attributes. The texton maps jointly define a conditional distribution over the pixels of the image. The nature of the model (in particular the fact that the number of textons and their neighborhood is not fixed but needs to be inferred as well) makes inference relatively expensive, and it is performed using reversible jump MCMC. Guo et al. (2007) propose a related model that is inspired by Marr's primal sketch. In this model, an image is first split into different two types of structure, a "sketchable" part, that is modeled in terms of contours, and a "non-sketchable" part that is modeled in terms of textured regions. The contours of the sketchable part are then composed from parameterized edge-elmements that are arranged into valid configurations by an inhomogeneous MRF similar to Guo et al. (2003).

The two major advantages of the more structured models discussed in this section are that they tend to account for particular aspects of natural images (e.g. contours) considerably more efficiently than generic models, and that they are directly formulated in terms of more abstract properties of an image and the representations therefore more interpretable (we have, for instance, latent variables that have prescribed roles in the model, and e.g. directly reflect the position of a particular edge-element). As a downside they are less flexible and much of the representation is not learned but rather imposed a priori by the model formulation.

2.3 Some approaches to approximate inference and learning

As discussed in section 2.1.2 above, exact inference and/or learning in many interesting models is intractable, and this is also true for most of the models that have been applied to image structure and which we have discussed in the preceding section. A broad range of different techniques for approximate inference and learning have been used in connection with these models. To overcome, for instance, the intractability of the posterior distribution in sparse coding models (section 2.2.2), deterministic approaches, such as using the posterior mode (e.g. Olshausen and Field, 1997), a Laplace approximation of the posterior (e.g. Lewicki and Olshausen, 1999), different variational approximations (e.g. Berkes et al., 2007; Girolami, 2001), and expectation propagation (e.g. Seeger, 2008) have been applied, but also sampling-based approaches (e.g. Olshausen and Millman, 2000). In the context of hierarchical models such as dynamic

trees or DBMs, variational inference with factorial distributions (mean field) has been used (e.g. Adams and Williams, 2003; Salakhutdinov and Hinton, 2009). Dayan et al. (1995), in the context of the Helmholtz Machine, consider a variational approach that uses an approximate posterior distribution that takes the form of a recognition network with weights independent of those of the generative model. Recognition networks have also been used more recently in the context of DBNs (Hinton et al., 2006b) and for DBMs to initialize mean-field inference (Salakhutdinov and Larochelle, 2010). For the models considered in this thesis, we will mainly use sampling based methods and we will discuss the two approaches for constructing Markov chains that we are mainly going to rely on in section 2.3.1.

All models that will be developed in the remainder of the thesis are at least partially undirected. This poses particular problems during learning due to the intractability of the normalization constant. We will discuss several techniques for learning in undirected models in section 2.3.2 below.

2.3.1 MCMC techniques

In this section we will discuss two MCMC techniques for simulating Markov chains that will be used in the remainder of the thesis: Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990; section 2.3.1.1) and Hybrid Monte Carlo (Neal, 1993, 2011; section 2.3.1.2). In section 2.3.1.3 we briefly discuss several issues that are frequently encountered when using MCMC techniques in practice.

2.3.1.1 Gibbs sampling

Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990) constructs a Markov chain by sampling (groups of) variables from the conditional distribution over these variables given the remaining variables. In the simplest case, one variable is sampled given the current values of the remaining variables in the model: $x_i \leftarrow x_i \sim p(x_i|\{x_j : j \neq i\})$. This is also referred to as single-site Gibbs sampling. The required one-dimensional conditional distribution can often be computed and sampled from exactly, or it can be efficiently approximated, and the update is applied to all variables in turn, in many cases in a fixed sequence but random choices are also possible. More general schemes jointly update sets of variables conditioned on the remaining ones and are referred to as "block" Gibbs sampling. Gibbs sampling can be thought of as applying a sequence of transition kernels $T = T_1 \dots T_N$, each of which changes only

one (or some) component(s) of the vector $\mathbf{x}$.

Although a Gibbs sampling scheme can be implemented in the majority of models, the effectiveness strongly depends on the model structure, in particular on the conditional independences and on the form of the conditional distributions involved. In some cases, suitable blocking strategies can lead to faster mixing of the chains and lower variance of the estimates obtained from the samples (e.g. Hamze and de Freitas, 2004). Other forms of blocking can lead to computationally very efficient samplers: For RBMs and other PoE models (e.g. Welling et al., 2003; Osindero et al., 2006), for instance, the conditional distributions over large sets of variables are factorial so that all hidden units can be sampled simultaneously given the visibles and vice versa (cf. equations 2.21, 2.22 above). Similarly, in DBMs (Salakhutdinov and Hinton, 2009; cf. also section 2.2.5), the units in one layer are conditionally independent given the layer directly above and below (as can be seen from the graphical model in Fig. 2.6c). Since the conditional distributions are also of convenient parametric forms (e.g. in the exponential family) it is possible to obtain parallelizable implementations of block Gibbs sampling that are computationally much more efficient than naïve single-site Gibbs sampling on commonly used computing architectures.

For other models, however, e.g. for many homogeneous MRF models, Gibbs sampling can be a lot less efficient, primarily due to a less favorable independence structure but also because the relevant conditional distributions do not necessarily have a convenient form. In some cases it can be advantageous to introduce additional (auxiliary) latent variables that then allow for efficient implementations of block Gibbs sampling. For instance, Schmidt et al. (2010) adopt for a variant of the FoE model (cf. chapter 3) an auxiliary variable approach similar to Welling et al. (2003); Osindero et al. (2006) and demonstrate improved results compared to the original HMC based approach in Roth and Black (2005). Martens and Sutskever (2010) describe a related strategy for general discrete pairwise MRFs.

2.3.1.2 Hybrid Monte Carlo

Simple forms of the Metropolis-Hastings algorithm often exhibit a random walk behavior leading to correlated samples and slow exploration of the state space. HMC (Duane et al., 1987; Neal, 1993, 2011), which we will use in chapter 3, aims to reduce this behavior in models with continuous random variables by generating more distant proposals using an analogy with physical systems: The energy that defines the distribution to be sampled from is considered as a "potential energy" and the variables over

which this distribution is defined specify the configuration of the system. In addition, a "momentum" is introduced into the system. The gradient of the energy then defines a force that acts on the configuration of the system and changes it via an interaction with the momentum in a deterministic manner. HMC further assumes an interaction of the system with a heat reservoir which is modeled as random. Combining the effects of the (deterministic) system dynamics and the random interactions with the external heat reservoir allow sampling from the distribution of interest in an efficient manner since the deterministic dynamics of HMC avoid the random walk behavior that is typically encountered with the naïve Metropolis algorithm.

Thus, to sample from the canonical distribution for a set of real variables $\mathbf{x} \in \mathbb{R}^N$ (e.g. the image pixels) defined in terms of an energy function $E(\mathbf{x})$ we extend the space by a second set of "momentum" variables, $\mathbf{m} \in \mathbb{R}^N$. These momentum-variables are independent and Gaussian distributed. The joint space of $\boldsymbol{X}$ and $\boldsymbol{M}$ is referred to as "phase space" and has a straightforward canonical distribution defined by the energy $H(\mathbf{x},\mathbf{m}) = E(\mathbf{x}) + K(\mathbf{m})$ where $K(\cdot)$ is the energy of a Gaussian distribution: $K(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{m}$. This joint energy is referred to as the Hamiltonian function. Using this function, one can now define a dynamics in the extended space, for which the x_i and m_i are considered as functions of time τ and we have

$$\frac{\mathrm{d}x_i}{\mathrm{d}\tau} = \frac{\partial H}{\partial m_i} \tag{2.31}$$

$$\frac{\mathrm{d}m_i}{\mathrm{d}\tau} = -\frac{\partial H}{\partial x_i}. \tag{2.32}$$

For each step of MCMC, a momentum is now drawn from the corresponding Gaussian distribution, the joint dynamics is then simulated for a certain period of time, and a sample of $\boldsymbol{X}$ is finally obtained by taking the end state of the trajectory and discarding the momenta. If the dynamics were simulated perfectly, then H would not change along a trajectory in phase space and the endpoint of a trajectory would always be accepted as a new sample. In practice, however, the dynamics need to be discretized using a non-zero time step ϵ. This introduces an error which leads H to change. To correct for this, trajectory endpoints are accepted with a probability that depends on the difference in energy between the start- and endpoint of a trajectory: Given an initial pair of values $(\mathbf{x},\mathbf{m})$ and a new pair after simulating the dynamics of the system $(\mathbf{x}^*,\mathbf{m}^*)$ the new pair is accepted with probability $\min(1, \exp\{-H(\mathbf{x}^*,\mathbf{m}^*) + H(\mathbf{x},\mathbf{m})\})$. Different discretization schemes can be used for approximately simulating the dynamics. One common choice is the leapfrog discretization which we also use in the experiments in chapter 3. A single iteration of this scheme computes $\mathbf{m}(\tau+\epsilon)$, $\mathbf{x}(\tau+\epsilon)$ at

time step $\tau + \epsilon$ from $\mathbf{m}(\tau)$, $\mathbf{x}(\tau)$ at time step τ by first performing a half-step with respect to m_i, then a full step with respect to x_i, and finally another half-step with respect to m_i:

$$
\begin{aligned}
m_i(\tau + \frac{\epsilon}{2}) &= m_i(\tau) - \frac{\epsilon}{2}\frac{\partial H}{\partial x_i}\left(x_i(\tau)\right) \\
x_i(\tau + \epsilon) &= m_i(\tau + \frac{\epsilon}{2}) + \epsilon\frac{\partial H}{\partial m_i}\left(m_i(\tau + \frac{\epsilon}{2})\right) \\
m_i(\tau + \frac{\epsilon}{2}) &= m_i(\tau) - \frac{\epsilon}{2}\frac{\partial H}{\partial x_i}\left(x_i(\tau + \epsilon)\right).
\end{aligned}
$$

See Neal (1993) for more details.

2.3.1.3 Issues with MCMC techniques

In practice, there are several difficulties associated with MCMC techniques. Although MCMC techniques are asymptotically exact, they are computationally very expensive. Unless a Markov chain is already initialized at its equilibrium distribution the samples in the initial portion of the chain are not samples from the equilibrium distribution. A significant fraction of the chain therefore often needs to be discarded, and determining the required length of this "burn in" period can pose additional difficulties. Furthermore, nearby samples are often highly correlated so that the effective number of *independent* samples is usually much lower than the raw number of samples obtained from a chain. One additional problem that is often encountered when sampling from multi-modal distributions in high dimensions is that the Markov chains do not mix well between modes since they have difficulties moving through the regions of low probability that separate the modes. This is indeed a problem also encountered for the models in this thesis. Various techniques have been proposed to overcome this problem, most of which rely on some form of "tempering", which involves sampling from a sequence of distributions at different temperatures, where transitions between modes is easier in the distributions at higher temperatures (e.g. Geyer, 1991; Marinari and Parisi, 1992; Neal, 1996). Efficient sampling techniques are especially important for approximate learning in complex undirected models where sampling from the model distribution is required to approximate the terms in the gradient arising from the intractable normalization constant (e.g. Desjardins et al., 2010; Salakhutdinov, 2010b,a).

2.3.2 Approximate learning in undirected graphical models

As discussed above, learning in undirected graphical models is problematic due to the fact that the normalization constant Z depends on the parameters but is generally not tractable to compute, nor is its gradient with respect to the parameters. Specifically, for a general undirected graphical model

$$p(\mathbf{x}_V;\Theta) = \frac{1}{Z(\Theta)} \int \mathrm{d}\mathbf{x}_H \tilde{p}(\mathbf{x}_V,\mathbf{x}_H;\Theta), \tag{2.33}$$

with parameters Θ, unobserved variables $\mathbf{x}_H$, unnormalized joint distribution $\tilde{p}(\mathbf{x}_V,\mathbf{x}_H) = \exp\{-E(\mathbf{x}_V,\mathbf{x}_H)\}$, and normalization constant $Z = Z(\Theta) = \int \mathrm{d}\mathbf{x}_V \mathrm{d}\mathbf{x}_H \tilde{p}(\mathbf{x}_V,\mathbf{x}_H)$, the gradient of the log-likelihood takes the following form:

$$\begin{aligned}\nabla_\Theta \log p(\mathbf{x}_V;\Theta) = &- \frac{\int \mathrm{d}\mathbf{x}_H \tilde{p}(\mathbf{x}_V,\mathbf{x}_H;\Theta)\nabla_\Theta E(\mathbf{x}_V,\mathbf{x}_H;\Theta)}{\int \mathrm{d}\mathbf{x}_H \tilde{p}(\mathbf{x}_V,\mathbf{x}_H;\Theta)} \\ &+ \frac{\int \mathrm{d}\mathbf{x}'_V \int \mathrm{d}\mathbf{x}_H \tilde{p}(\mathbf{x}'_V,\mathbf{x}_H;\Theta)\nabla_\Theta E(\mathbf{x}'_V,\mathbf{x}_H)}{Z} \qquad (2.34) \\ = &- \left\langle \nabla_\Theta E(\mathbf{x}_V,\mathbf{x}_H;\Theta) \right\rangle_{p(\mathbf{x}_H|\mathbf{x}_V)} + \left\langle \nabla_\Theta E(\mathbf{x}_V,\mathbf{x}_H;\Theta) \right\rangle_{p(\mathbf{x}_V,\mathbf{x}_H)}. \qquad (2.35)\end{aligned}$$

The first term is the expectation of the gradient of the energy with respect to the posterior distribution of the latent variables given the observed data. As discussed above, for models such as the RBM this term is tractable, and for fully observed models such as the MRF considered in chapter 3 it is trivial. The second term, however is problematic in most models. It arises from the derivative of the log-normalization constant with respect to the parameters and involves an expectation with respect to the current model distribution which typically cannot be computed.

Several approaches have been brought forward that deal with this problem and two broad strategies can be distinguished: The first one is to approximate the intractable term using either sample-based or deterministic approaches. The second one is to choose an alternative learning – or inductive – criterion that avoids computing this term altogether. Below we will discuss examples from both groups. A recent review and experimental evaluation can be found e.g. in Marlin et al. (2010), who provide a comparison in the context of RBM learning.

2.3.2.1 Alternative inductive criteria

Examples of alternative inductive criteria include maximum pseudo-likelihood, score matching, noise contrastive estimation, or contrastive divergence. Whereas the former

two are deterministic criteria, the latter two are based on sampling techniques. Maximum pseudo-likelihood (Besag, 1974) and its generalizations, composite likelihood (e.g. Lindsay, 1988 and more recent work; see also Vickrey et al., 2010 for a related approach), can be thought of as an attempt to optimize the parameters of the model such that certain conditional distributions of the model match those of the data. Another example in this group is score matching (Hyvärinen, 2005) with its extensions (Hyvärinen, 2007) which attempts to match the score function (gradient of the log-density) of the model distribution.

Noise contrastive estimation (Gutmann and Hyvrinen, 2010; Pihlaja et al., 2010) uses what is effectively a discriminative approach: It treats the normalization constant Z as an additional model parameter and estimates it jointly with the true parameters by training a logistic classifier to distinguish between samples from the data distribution and samples from some noise distribution.

Contrastive divergence (CD; Hinton, 2002) is an alternative learning criterion that uses a sample based approximation of the second term, however, the samples are not samples from the true model distribution, but rather samples from a distribution that is obtained by running a Markov chain for T steps using a transition operator that leaves the model distribution invariant (e.g. Gibbs sampling for RBMs or HMC for continuous valued MRFs, see below) starting at the data distribution. If $T \to \infty$ the Markov chain will converge to the model distribution and approximate (stochastic) maximum likelihood learning is recovered. Typically, however, T is chosen to be small (often $T = 1$, in most cases $T < 10$). CD does not approximate the true likelihood gradient, but it approximately minimizes the difference between two divergences (Hinton, 2002): $\mathrm{KL}\left[P^0||P^\infty_\Theta\right] - \mathrm{KL}\left[P^T_\Theta||P^\infty_\Theta\right]$, where P^0 is the data distribution, P^∞_Θ the model distribution, and P^T_Θ is the distribution obtained by initializing a Markov chain at the data and then simulating it for T steps (note that the first term will always be larger than the second, unless the distribution defined by the model is equal to the data distribution). CD has been extensively used for training RBMs (e.g. Hinton et al., 2006b) but also for other undirected models (e.g. Roth and Black, 2005 and Hinton et al., 2006a). Its properties have been studied by various authors (Yuille, 2004; Carreira-Perpiñ and Hinton, 2005; Bengio and Delalleau, 2009; Sutskever and Tieleman, 2010), and it has been found to often work reasonably well in practice although its effectiveness can depend strongly on the chosen T as well as on nature of the model and the data distribution to be fitted. One major limitation of CD (especially for small T) is that the brief chains will never move far away from the data. Intuitively speaking, this makes

the CD update largely agnostic to the shape of the learned distribution in regions of the space where no data is observed, and it can have difficulties correctly estimating the relative mass of well isolated modes in the data distribution (Hinton et al., 2003 see also chapter 3 for a related discussion). While this is less of a problem when CD is used to learn representations ("features") that are subsequently used in a discriminative scheme, it can be problematic if the goal is to estimate density models, which can be poor, especially for CD-1, and strongly depend on T (see e.g. Salakhutdinov and Murray, 2008 for a quantiative results in the context of RBMs). Unfortunately, other alternative criteria such as maximum pseudo-likelihood or variants of score matching are likely to suffer from similar "locality"-issues (e.g. Marlin et al., 2010; Vickrey et al., 2010). One potential advantage of CD is that it deals with latent variables more naturally than most of the other alternative criteria discussed above.

2.3.2.2 Approximations to the true likelihood gradient

Approximations of the intractable term in the likelihood gradient can be obtained using several of the deterministic approaches for approximate inference, such as variational techniques or loopy belief propagation briefly discussed in section 2.1.2. The distribution that needs to be approximated, however, is likely to be more complicated than e.g. a posterior distribution over hidden variables (for instance, it is less likely to be well described by a single mode) and the approximation therefore likely to be worse. Monte Carlo approximations obtained by sampling from the model distribution using e.g. importance sampling or MCMC (e.g. Geyer and Thompson, 1992) are also conceivable but can be problematic as the resulting gradient estimates are likely to suffer from high variance, and they can also be very slow since they require to repeatedly run Markov Chains to convergence as the model parameters change.

Recently, a class of efficient sampling-based approaches have re-gained popularity. Younes (1989) showed that it is not necessary to run a Markov chain to convergence after each update of the model parameters. Instead, it is possible to alternate between single updates of the Markov chain and updates of the model parameters using the gradient computed from the current samples in the chain(s). This approach has recently been adopted by Tieleman (2008) as an alternative to CD for training RBMs and it is now also widely used. This technique is a stochastic approximation method and often referred to as "persistent" CD (PCD)[2] although "stochastic maximum likelihood"

[2]The term "persistent CD" results from the fact that in practice the method is implemented in a manner very similar to standard CD, the only difference being that the Markov chains for the second

(SML) is more appropriate. The intuition is that each parameter update will only lead to relatively small changes of the model distribution so that by alternating parameter updates and sampling steps the particles in the persistent chains remain representative of the model distribution as its parameters change and can thus be used to compute the required expectations. Compared to CD the persistent chains are much more likely to explore the full model distribution. One important pre-requisite for this is, however, that good mixing of the persistent chains is ensured. This can be difficult especially late in learning. Several of the techniques for improving the mixing of chains discussed in section 2.3.1.3 have been employed in the context of SML (e.g. Desjardins et al., 2010; Salakhutdinov, 2010b,a), but more specialized approaches have also been developed (e.g. Tieleman and Hinton, 2009).

2.3.2.3 Practical considerations and learning with stochastic gradient

In this thesis, we will primarily use CD and SML (persistent CD). Both will be used in the context of stochastic gradient ascent (SGA; e.g. Bottou, 2004). For SGA the data is split into several *mini-batches* that are then processed in sequence, and a gradient update is computed and applied after each mini-batch. A full sweep through the training data (i.e. all mini-batches) is often referred to as *epoch*. This approach can be computationally advantageous to computing updates from the full dataset since it allows for more gradient steps per unit of time. This applies in particular in situations when computing the gradient from a data point is time consuming, which is, for instance, the case when expensive inference is required as in chapters 4 and 5. Further details of the implementations of these methods in the context of different models will be provided in later chapters. General practical guidance to the use of CD, SML, and SGA can also be found, for instance, in Hinton (2010), who discusses their use in the context of RBMs.

term are not initialized at the data but instead with the samples after the previous iteration.

Chapter 3

Learning generative texture models with extended Fields-of-Experts

3.1 Introduction

In chapter 1 we have argued for a more structured approach to modeling low- and mid-level structure in images. In this chapter we will motivate this general idea further by investigating the power of probabilistic prior models of *generic* image structure. Such models of generic structure are important for many image processing and synthesis tasks, and suggest themselves as building blocks of more comprehensive probabilistic models of natural scenes. However, as argued in chapter 1, natural images are extremely complex and typically contain different regions with very different visual characteristics. Attempting to learn these different characteristics jointly with a relatively simple model might not be possible and a likely outcome is that the model ends up capturing only very basic properties such as the piecewise smoothness of images. While this might be sufficient for certain image processing tasks (such as denoising or simple inpainting), it is not very satisfying in terms of more general models of natural image structure. As discussed in chapter 1 it might therefore be more appropriate to focus on models that are good at capturing the visual characteristics of *individual* regions (e.g. specific textures), and then use these kinds of models as building blocks of more comprehensive, hierarchical formulations that can account e.g. for images comprised of multiple regions. With this in mind we will attempt to better understand the generative power of existing models of generic image structure. In particular, we will attempt to answer the question whether a model of generic image structure is also likely to be a good model of specific image structure, and, if not, what the important properties of

models of specific image structure are.

We will focus on one successful example of a generic image prior, the Field-of-Experts (FoE) framework, recently proposed by Roth and Black (2005). The FoE defines a probability distribution over images in the form of a homogeneous high-order Markov random field (MRF) the clique potentials of which are defined in terms of the responses of linear filters. This MRF-based model is translation invariant and can be applied to images of arbitrary size. It is fully parametric and all parameters can be learned from training data. Thus it can be directly adapted to the statistics of natural images.

Several studies have demonstrated that the FoE performs very well in tasks which require a generic image prior, such as image denoising, inpainting and novel view synthesis (Roth and Black, 2005; McAuley et al., 2006; Woodford et al., 2006). However, while the FoE's suitability for certain tasks is certainly encouraging, other results, such as the very smooth nature of samples drawn from a FoE model trained on natural image patches (Roth, 2007, Fig. 4.9) and e.g. the analyses of closely related models (Weiss and Freeman, 2007; Tappen, 2007; c.f. section 3.3.1 below) suggest that the FoE might still be a relatively limited model of natural images and accounts predominantly for their piecewise smoothness.

In this chapter we pursue this idea further and evaluate the FoE's generative power. In order to obtain more insight as to what kind of structure can be modeled by the FoE we focus on one particular test case: modeling synthetic and natural image textures. The motivation for this is two-fold. Firstly, evaluating an undirected graphical model is generally difficult, since a quantitative assessment e.g. by computing the log-likelihood is not possible (or at least computationally very expensive; see discussion in section 2.1.1.2 of chapter 2). One useful alternative that has been applied in the literature is to draw samples from trained models and compare them to the training data. Unfortunately though, for generic image priors it is not clear what kind of "generic" structure such a model can be expected to learn. This makes it hard to assess the quality of the samples and thus reduces the usefulness of the approach. The focus on specific image structures appears to be an interesting alternative: Here we have very clear expectations as to what samples should look like and a failure of the model to account for the structure allows us to investigate more directly in what respect the model is lacking. Secondly, in the context of our longer-term goal to develop more powerful, structured image models a good model of image texture is obviously of great importance. Unfortunately, however, many of the most powerful methods for generating specific structure

that have been proposed in the past are not formulated as probabilistic models and it is therefore not clear how they could be used in this context (cf. section 3.3.2).

Interestingly, we find that the FoE in its original form is limited and not able to model individual image textures. We show that it does not perform better than the much simpler Gaussian FoE on this task and we provide an intuitive explanation as to why this is the case. In order to make progress towards our goal of developing more structured models of natural images we then devise a modification of the FoE and demonstrate how changing the structure of the model and using a more powerful learning algorithm can substantially increase the generative power, giving rise to a compact parametric model of natural textures that can be fully learned from training data and the performance of which is comparable with state-of-the-art nonparametric approaches such as (Efros and Leung, 1999) in the experiments shown. This model will be the basis of a more comprehensive model for images with multiple texture regions that we will develop in section 3.5.

The rest of the chapter is structured as follows: In section 3.2 we describe in detail the FoE as proposed by Roth and Black (2005) and our extension that allows modeling individual visual textures. Related work will be discussed in section 3.3. Section 3.4 assesses the generative power of the FoE and compares it with our extended model on various texture modeling tasks in 1D and 2D. We provide some insight into the large differences in performance of the two models in section 3.4.6. Section 3.5 illustrates how the model could be used as a component of a more comprehensive, region-based model of images. We conclude with a discussion in section 3.6.

3.2 Models

Below we first explain the FoE as proposed by Roth and Black (2005); Roth (2007) (section 3.2.1) and some variants obtained by using different potential functions. Section 3.2.2.3 then describes the extended model with bimodal potentials. Section 3.2.3 finally discusses how the different variants of the model can be learned.

3.2.1 Field of Experts

The FoE as proposed by Roth and Black (2005) (see also Roth, 2007, for a more detailed exposition) defines a probability density over continuous-valued images. It is motivated by the need for (generative) prior models of low-level structure in natural im-

ages. It incorporates insights derived from the study of the statistics of natural images, in particular the heavy-tailed nature of filter response marginals and the long-range dependencies between image pixels, which have also been incorporated into many generative models of image *patches* (Olshausen and Field, 1997; Bell and Sejnowski, 1997; Welling et al., 2003; see also sections 2.2.2 – 2.2.4 in the previous chapter). Yet, unlike these models of patches, the FoE is a stationary MRF (cf. section 2.2.1 in chapter 2) and it is thus translation invariant and can be applied to images of arbitrary size. Thanks to its particular parametric form it can be fully learned from data.

A good way to understand the FoE is by starting with the Products-of-Experts (PoE) model (Hinton, 2002; Welling et al., 2003). The FoE can be thought of as an extension of that model. As discussed in section 2.2.3 of the previous chapter, in the PoE framework high-dimensional probability distributions are modeled by taking the product of several distributions (the experts), each of which may be defined on a lower-dimensional subspace of the data, and in the case of images, a one-dimensional subspace is typically used (e.g. Welling et al., 2003). The general formulation of this model is given in equation (2.16) of section 2.2.3 and reproduced here for completeness:

Considering an image $\mathbf{x}$ as a vector of length N, i.e. $\mathbf{x} \in \mathbb{R}^N$, each expert distribution is defined in terms of $\mathbf{w}_j^T \mathbf{x}$ where $\mathbf{w}_j$ defines the subspace of expert $j = 1 \dots M$ (where M is the number of experts). Thus

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{j=1}^{M} \Phi(\mathbf{w}_j^T \mathbf{x}; \boldsymbol{\alpha}_j) \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the image, $\mathbf{w}_j$ defines the subspace of expert $j = 1 \dots M$, $\Phi(y; \boldsymbol{\alpha}_j)$ is an nonlinear expert function with parameters $\boldsymbol{\alpha}_j$ (typically an unnormalized 1D density function), and Θ is the set of parameters of the model (basis vectors $\mathbf{w}_j$s and expert parameters $\boldsymbol{\alpha}_j$s); $Z(\Theta) = \int \prod_{j=1}^{M} \Phi(\mathbf{w}_j^T \mathbf{x}; \boldsymbol{\alpha}_j) \mathrm{d}\mathbf{x}$ is the normalization constant.

In the PoE, $\mathbf{w}_j$ has the same size as the image and a PoE is therefore typically limited to small images (image patches). Directly applying this model to larger images would have several drawbacks: it would be computationally very expensive, the number of parameters be extremely large, there would be no translation invariance (a desirable property for a prior model of low-level image structure), and any PoE trained on images of a particular size would only be valid for images of that size.

These shortcomings are addressed by the FoE. Here, the $\mathbf{w}_j$s are much smaller than the images of interest (e.g. 5×5 pixels in Roth and Black, 2005) but the experts are

replicated at each pixel. This allows the application of FoEs to images of arbitrary size while keeping the number of parameters low:

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{i=1}^{N} \prod_{j=1}^{M} \Phi(\mathbf{w}_j^T \mathbf{x}_{(i)}; \boldsymbol{\alpha}_j), \tag{3.2}$$

Here the index i runs over the pixels in the image, and $\mathbf{x}_{(i)}$ is the image patch of the same size as $\mathbf{w}_j$ centered at pixel i.

The fact that the experts are replicated at each pixel and applied to the local image patch gives rise to an appealing interpretation of the FoE: The $\mathbf{w}_j$s effectively act as linear filters and the FoE is thus a homogeneous high-order MRF with clique potentials defined in terms of the responses of these linear filters

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{i=1}^{N} \prod_{j=1}^{M} \Phi(y_{ji}; \boldsymbol{\alpha}_j), \tag{3.3}$$

where $y_{ji} = [x * \bar{w}_j]_i$ i.e. the y_{ji} is the response of filter $\bar{w}_j$ at pixel i ($*$ denotes convolution and $\bar{w}_j$ is a flipped version of $\mathbf{w}_j$ to account for the convolution). The size of the cliques of this MRF is determined by the size of the filters $\mathbf{w}_j$, there is one clique centered at each pixel, and the potentials of these cliques are given by the product of all one-dimensional experts centered at the respective pixel.

In the PoE case, for $\mathbf{x} \in \mathbb{R}^N$, we need $M \geqslant N$ for the model to define a valid density over $\mathbb{R}^N$ (intuitively speaking, $\mathbf{x}$ would otherwise remain unconstrained in some direction). In the FoE, however, since the "expert" is replicated at every pixel $M = 1$ can be sufficient[1]. In practice, however, M is chosen considerably larger (e.g. Roth and Black (2005) choose $M = 24$), so that the model is highly overcomplete since the effective number of experts NM is much larger than N, the dimensionality of the image.

While the interpretation of the FoE as a PoE replicated at every pixel in the image is appealing this view glosses over one critical aspect of the FoE which is highlighted by the normalization constant:

$$Z(\Theta) = \int \prod_{i=1}^{N} \prod_{j=1}^{M} \Phi(\mathbf{w}_j^T \mathbf{x}_{(i)}; \boldsymbol{\alpha}_j) \mathrm{d}\mathbf{x} \tag{3.4}$$

This normalization constant is not simply a product of the normalization constants corresponding to the PoEs applied at each pixel. Instead, it takes the interactions between PoEs located at nearby pixels into account. Thus, the parameters of a FoE cannot be learned by simply learning parameters for a PoE on image patches of the same size

[1] Although the image boundaries might cause problems in practice.

as the filters $\mathbf{w}_j$ and then re-normalizing. Instead, through the normalization constant in eq. (3.4) these interactions are taken into account during learning (see also section 3.2.3 below).

As explained in section 2.1.1.2 of chapter 2 it is convenient to write the density in terms of an energy $E(\mathbf{x};\Theta)$, so that the density is then given by the corresponding Gibbs distribution:

$$E(\mathbf{x};\Theta) = -\sum_{i=1}^{N}\sum_{j=1}^{M}\Psi(\mathbf{w}_j^T\mathbf{x}_{(i)};\boldsymbol{\alpha}_j) \tag{3.5}$$

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)}\exp(-E(\mathbf{x};\Theta)), \tag{3.6}$$

with $\Psi(\cdot;\boldsymbol{\alpha}) = \log\Phi(\cdot;\boldsymbol{\alpha})$ (see also equations 2.3 and 2.4).

3.2.2 The choice of expert function

Different choices are possible for the expert function $\Phi(y;\boldsymbol{\alpha})$ and as we will demonstrate below, they give rise to models with very different characteristics. For the learning and sampling strategies below it is required that the potential function and its logarithm are continuous and differentiable with respect to y and to the parameters $\boldsymbol{\alpha}$.

Commonly $\Phi(y;\boldsymbol{\alpha})$ is chosen such as to be a valid unnormalized one-dimensional density. This is not strictly required but it ensures that the model always defines a valid density over $\mathbb{R}^N$ (choosing $\Phi(y;\boldsymbol{\alpha})$ to be a valid one-dimensional density guarantees that the full model can be normalized; however, since the FoE is overcomplete when $M > 1$ the full model might be normalizable even when individual experts do not define valid one-dimensional densities themselves).

3.2.2.1 Student-t potential

Roth (2007) investigates two heavy-tailed (kurtotic) expert functions, the unnormalized, one-dimensional Student-t density and a smooth approximation to the L1 norm which is referred to as the Charbonnier expert (the latter is slightly less heavy-tailed than the Student-t expert). In most applications Student-t experts have been used (e.g. Roth and Black, 2005) and this is what we will focus on here. The unnormalized Student-t density is given as follows:

$$f_t(y;\nu,\sigma,\mu) \propto \left(1 + \frac{1}{\nu}\left(\frac{y-\mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}. \tag{3.7}$$

Roth and Black (2005) assume μ to be zero (an assumption that will be discussed in more detail below). Since σ can be absorbed into $\mathbf{w}$ they simplify the expert function as follows:

$$\Phi(y;\nu) = (1+\frac{1}{2}y^2)^{-\nu} \tag{3.8}$$

with $\nu > 0$. Thus, (3.2) can be written in terms of the energy as:

$$p_{FoE}(\mathbf{x}) = \frac{1}{Z(\Theta)}\exp(-E_{FoE}(\mathbf{x})); \tag{3.9}$$

$$E_{FoE}(\mathbf{x}) = \sum_i\sum_j v_j \log\left\{1+\frac{1}{2}\left(\mathbf{w}_j^T\mathbf{x}_{(i)}\right)^2\right\}. \tag{3.10}$$

The choice of heavy-tailed experts, in particular of the Student-t potential, is motivated by the fact that responses of linear filters to natural images typically exhibit highly kurtotic response distributions. Student-t potentials have also been used e.g. by Welling et al. (2003) in their Product-of-Experts model of image patches and sparsity priors have been proven to be effective in directed models of image patches (e.g. Olshausen and Field, 1997; Bell and Sejnowski, 1997; see discussion in section 2.2.2).

3.2.2.2 Squared exponential

Choosing the expert function to be a squared exponential, i.e.

$$\Phi(y) = \exp\{-\frac{1}{2}(y+b)^2\}, \tag{3.11}$$

gives rise to the structurally similar but qualitatively quite different generative model, the Gaussian Field of Experts (GFoE). The GFoE has energy

$$E_{GFoE}(\mathbf{x}) = \frac{1}{2}\sum_i\sum_j\left(\mathbf{w}_j^T\mathbf{x}_{(i)}+b_j\right)^2 \tag{3.12}$$

and defines a Gaussian distribution over the set of images, i.e.

$$\mathbf{x} \sim N(\boldsymbol{\mu},\Sigma) \tag{3.13}$$

with

$$\boldsymbol{\mu} = -(\sum_j W_j^T W_j)^{-1}(\sum_j W_j^T \mathbf{1} b_j) \tag{3.14}$$

$$\Sigma = (\sum_j W_j^T W_j)^{-1}. \tag{3.15}$$

Here, W_j is the convolution matrix corresponding to $\mathbf{w}_j$ and $\mathbf{1}$ is a vector of ones.

The GFoE is a Gaussian MRF (GMRF) model; these have been used for texture modeling for many years, see e.g. (Chellappa et al., 1985). In particular, unlike for the FoE described above, as a Gaussian distribution the model is analytically tractable in that the normalization constant Z and thus also the likelihood of the GFoE and its gradient can be computed exactly.

Although the GFoE is structurally similar to the FoE it is well known that it models only the power spectrum of an image and not the phases so its ability to capture image structure is very limited. However, the GFoE has recently served as the basis of studies that aim at understanding the computational properties of the Student-t FoE (Weiss and Freeman, 2007; Schmidt et al., 2010; see also section 3.3), and we will use it as a baseline model in our experiments below.

3.2.2.3 Extended FoE with bimodal potentials

In our experiments we find that the FoE in the form presented by Roth and Black (2005) is not able to model natural textures or even simple one-dimensional periodic patterns (cf. section 3.4). One possible explanation is that the particular expert function used by Roth & Black is too restrictive. Indeed, as will be discussed in more detail in section 3.4.6 the Student-t FoE is unimodal and the findings in section 3.4 suggest that this might severely affect the generative power of the model.

We therefore propose an extension of the original FoE model which allows for bimodal expert functions

$$\Phi_{Bi}(y;v,a,b) = \left\{1 + \frac{1}{2}\left[(y+b)^2 + a\right]^2\right\}^{-v}. \tag{3.16}$$

This choice of Φ gives rise to the following energy:

$$E_{Bi}(\mathbf{x}) = \sum_i \sum_j v_j \log\left\{1 + \frac{1}{2}\left[\left(\mathbf{w}_j^T \mathbf{x}_{(i)} + b_j\right)^2 + a_j\right]^2\right\}.$$

Figure 3.1b shows the shape of the potential for different settings of the parameters a and b. b determines the overall position of the expert function along the y-axis while a controls its shape: Φ_{Bi} is unimodal for $a > 0$ but bimodal for $a < 0$.

As we will demonstrate in section 3.4 and discuss in more detail in 3.4.6 including the two additional parameters gives rise to a considerably more powerful model while the general principles of learning and inference (discussed the below) remain the same.

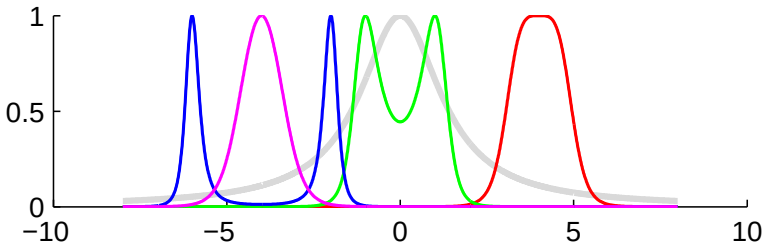


Figure 3.1: **Different BiFoE potentials**. Red: $a = 0$, $b = -4$; Green: $a = -1$, $b = 0$; Blue: $a = -4$, $b = -4$; Magenta: $a = 1$, $b = 4$. Gray: Student-t potential as used in the FoE. $v = 1$ in all cases.

3.2.3 Learning in the FoE

As explained in section 2.3.2 of chapter 2 the gradient of the log likelihood of the FoE with respect to the parameters $\boldsymbol{\alpha}_j$ and $\mathbf{w}_j$ (generically denoted as θ_j below) is given by

$$\frac{\partial \log p(\mathbf{x};\Theta)}{\partial \theta_j} = -\left\langle \frac{\partial E_{FoE}(\mathbf{x};\Theta)}{\partial \theta_j} \right\rangle^{+} + \left\langle \frac{\partial E_{FoE}(\mathbf{x};\Theta)}{\partial \theta_j} \right\rangle^{-} \tag{3.17}$$

and consists of two terms: the expectation of the gradient of the energy over the data distribution $\langle\cdot\rangle^{+}$ as well as over the model distribution $\langle\cdot\rangle^{-}$ (given the current model parameters Θ).

For the FoE models discussed in the previous section 3.2 the first term is trivial to compute, but the second term is intractable so that exact maximum likelihood learning is not possible. In chapter 2.3.2 we have outlined several approximations and alternative inductive critertia. Roth & Black propose learning the FoE using contrastive divergence (CD; Hinton, 2002; Hinton et al., 2006a). CD approximates the gradient by replacing $p_{FoE}(\mathbf{x};\Theta)$ with the distribution $\tilde{p}_T(\mathbf{x};\Theta)$ which is obtained by initializing the Markov chain at the data and running MCMC for only a small number of steps T. Roth & Black chose $T = 1$ in their experiments. As discussed in chapter 2, however, CD with a small T can be problematic and in our experiments we indeed found that CD-1 was not sufficient to obtain good estimates of the parameters for the BiFoE model. We therefore used stochastic maximum likelihood (SML; see chapter 2.3.2). Here, the sample based approximation of the second term of the gradient is obtained by using the samples from *K persistent* Markov chains that are initialized at the beginning of learning and updated in alternation with the model parameters. For the GFoE we compute the exact gradient of the log-likelihood with respect to the model parameters and perform stochastic gradient ascent.

3.2.4 Sampling from the FoE

Learning, and also texture generation or inpainting (see section 3.4) requires the ability to draw samples from a given FoE-model.

For the GFoE and images of moderate size this is can easily be done by computing the mean and covariance matrix of the corresponding Gaussian according to equations (3.14, 3.15) and then using conventional sampling techniques (e.g. computing the Cholesky-decomposition of the covariance matrix to transform samples generated from $N(\mathbf{0}, \boldsymbol{I})$).

For the Student-t FoE and the BiFoE direct sampling is not possible and one needs to resort to MCMC techniques. Naïve Gibbs sampling (cf. section 2.3.1.1) where one x_i is sampled at a time conditioned on all remaining pixels is a possibility but rather inefficient. For the Student-t FoE several efficient sampling techniques are available. One possibility is to use an auxiliary variable Gibbs sampler proposed for instance for the Student-t PoE in Welling et al. (2003) and more recently also applied to a variant of the FoE (Schmidt et al., 2010). Such an auxiliary variable sampler can however not easily be constructed for the BiFoE and we therefore follow the approach taken by Roth and Black (2005) (see also Hinton et al., 2006a) and use a Hybrid Monte Carlo (HMC) sampler (see discussion in section 2.3.1) which can be applied to the Student-t FoE as well as to the BiFoE.

3.3 Related Work

A general review of probabilistic generative models (including MRFs and other undirected models) for low- and mid-level vision has already been given in Chapter 2. A detailed discussion of the relationship of the FoE to many of these models can also be found in the original work by Roth (Roth, 2007; sections 2, 3, and 4). In this section we will instead briefly discuss two lines of work directly related to the goals of this chapter. Firstly, as we are interested in gaining a better understanding of representational power of the FoE model we will discuss in more detail several studies that have recently attempted to shed some light onto the computational properties of the FoE (section 3.3.1). Secondly, since we are further interested in developing probabilistic, parametric models of image texture that can be used as components of more comprehensive image models we will briefly discuss related work on texture synthesis in the computer vision literature (section 3.3.2).

3.3.1 Understanding the computational properties of the FoE

The Student-t FoE proposed by Roth and Black (2005) was designed as a generative model of low-level image structure and it has been applied very successfully to a range of image processing tasks.

Despite its success in applications, however, the FoE has proven hard to analyze and the underlying computational mechanisms have remained elusive for some time. In particular, the apparently random filters learned by the model have appeared somewhat counterintuitive. Also, the approximate learning criterion (contrastive divergence) together with the necessity to manually modify some of the learned parameters to achieve good application performance has raised questions as to how the model's application performance related to its quality as a generative model of natural images. The intractability of the normalization constant has made it difficult to answer such questions, since it is, for instance, not possible to compute the likelihood under the model which makes the direct comparison of a given FoE with other models (in particular with a FoE with different filters) very difficult. Recently, however, two noteworthy studies have made progress on the above questions by analyzing closely related models:

In an insightful study, Weiss and Freeman (2007) build on the Gaussian case in an attempt to understand the nature of the filters learned by the FoE. They start by analyzing a Gaussian PoE and demonstrate that here the expected filters correspond to the minor components of the data (Williams and Agakov, 2002) and are thus expected to have high-frequency content. Extending their analysis to the translation invariant GFoE they find that in this case the preferred filter should furthermore approximately satisfy the tiling constraint; in the simplest case of a model with a single expert the optimal $\mathbf{w}$ is simply a whitening filter. They finally analyze a FoE in which they replace the Student-t potential with a Gaussian scale mixture, made up of a finite number of Gaussians at different scales (a similar formulation has recently also been used in Schmidt et al., 2010). This allows them to formulate upper and lower bounds on the partition function and thus to compare different filters. They find that here, too, one would expect the model to learn filters that that fire rarely on natural images and frequently on all other images – which is achieved by filters that have large high-frequency content, as is indeed the case for the filters learned by Roth and Black (2005). This can be interpreted in terms of a smoothness constraint imposed by the model.

Along similar lines Tappen (2007) demonstrated the similarity between the FoE and a MRF designed around a fixed set of derivative filters. The latter can be expressed

as an outlier process (Black and Rangarajan, 1996) which can be viewed as imposing high-order piecewise continuity on the estimated image. The similarity between the Student-t FoE and the model based on derivative filters suggests that the underlying computational mechanisms are related and that the FoE can thus also be seen as imposing similar piecewise higher-order continuity on images. The results of Weiss and Freeman (2007) and Tappen (2007) are both are consistent with the smooth nature of the samples that are generated by a FoE trained on natural images (Roth, 2007; Fig. 4.7), supporting the idea that the FoE primarily models smoothness constraints with a robust loss function.

Piecewise smoothness is certainly an important characteristic of natural images, yet it is also a rather basic one and one might wonder why the model does not account for more advanced properties of natural images. In particular, one might wonder whether this is a limitation inherent to the model formulation, of the particular configuration investigated (e.g. filter size, number of experts), or possibly of the way the model has been trained (approximate learning with CD)?

One indication that the Student-t potential might not be sufficiently flexible in the general case is given by two studies Zhu and Mumford (1997) and Zhu et al. (1998) who have proposed a a model fundamentally very similar to the FoE which is called "FRAME". The FRAME model is also a homogeneous MRF with clique potentials defined in terms of the responses of linear filters. However, unlike the FoE, it is based on fixed filters while the potential functions are non-parametric. It is effectively the maximum entropy model with the histograms of filter responses as sufficient statistics. During learning filters are selected in an iterative fashion from a pre-specified filter bank and included in the model. Given a set of filters the model learns one parameter for each histogram bin for each filter, i.e. the expert functions ϕ introduced above are effectively modeled in a "non-parametric" manner as piecewise constant functions (constant across the width of a histogram bin), but with no other constraints imposed. The model can be trained on arbitrary sets of images and achieves good performance e.g. on visual textures and also for capturing basic image statistics. A disadvantage of the FRAME model compared to the FoE is the fact that due to the representation of the expert nonlinearities the filters themselves cannot be learned and the model is not amenable to efficient HMC sampling. Also, its effectively "non-parametric" (histogram-based) formulation makes it potentially harder to incorporate in a hierarchical framework. At the same time, however, the histogram-based potentials allow for considerably more flexibility and it is interesting to note that Zhu and coworkers find

that symmetric, decaying potentials (such as the Student-t potential) are not sufficient in order to obtain good generative models of natural images and textures, raising the question of how the choice of a particular parametric form for the potential function (such as the Student-t potential) affects the expressiveness of the FoE. (In fact, even when limiting oneself to symmetric, zero-centered expert functions Student-t densities appear not to be the best choice as has recently been proposed by Schmidt et al. (2010), instead even more heavy tailed distributions should be preferred.)

3.3.2 Generative models of image texture

A large body in the computer vision literature is dedicated to the problem of synthesizing visual textures. Some work has modeled the actual physical processes underlying the generation of natural textures (e.g. Witkin and Kass, 1991; Worley, 1996; Dorsey et al., 1999). A larger class, however, has focused on the statistical characterization of textures. This second approach can be traced back to Julesz (1962) who suggested that the Nth-order joint empirical densities of image pixels could be used characterize textures (in the sense that textures identical with respect to these statistics should be pre-attentively indistinguishable to human observers).

Two related lines of work on stochastic texture modeling can be found in the literature: Firstly, A large number of approaches attempt to model textures by directly formulating an MRF from which new instances of the texture can then be generated by sampling. One example of this type of model is the FRAME model (Zhu et al., 1998) discussed in the previous section, but many other MRF models have been proposed (e.g. Hassner and Sklansky, 1980; Cross and Jain, 1983; Chellappa et al., 1985; Derin and Elliott, 1987; Gimel'farb, 1996; Paget and Longstaff, 1998; Efros and Leung, 1999; Zalesny and Gool, 2001). A second line of work represents textures in terms of constraints defined on feature functions (e.g. Heeger and Bergen, 1995; De Bonet, 1997; Portilla and Simoncelli, 2000). The constraints are typically formulated in terms of the responses of linear filters which are, however, not learned but pre-specified. New instances of a texture are generated by creating an image satisfying the constraints. For instance, Portilla and Simoncelli (2000) start off with a random noise image and repeatedly apply transformations such as to obtain a projection of this random image onto the constraint surface. Although there are some connections to e.g. maximum entropy modeling (Portilla and Simoncelli, 2000) in general these approaches do generally not define explicit probabilistic models.

Most MRF models of textures use only first and second-order terms but ignore higher-order information even though Julesz has demonstrated that second-order information is not sufficient to characterize textures (but see see e.g. Zhu et al., 1998 for a high-order model). Furthermore it is interesting to note that models in both group are almost exclusively formulated in an at least partially non-parametric fashion (with Portilla and Simoncelli, 2000 being an example): As discussed above, for instance, Zhu et al. (1998) use a histogram-based representation of the filter marginals, while other models use histograms to represent the gray-level differences between pixels (e.g. Gimel'farb, 1996; Zalesny and Gool, 2001). In fact, Efros and Leung (1999), who demonstrate excellent results for a broad range of textures, side-step the process of constructing a model completely and simply "query" an example texture-image at each step during the synthesis of a novel texture instance: In their approach, a novel texture image is synthesized by growing one pixel at a time starting from some existing seed region. In each step, the already synthesized neighborhood of the pixel is compared with the sample texture image and the value for the pixel under consideration is then chosen according to the distribution defined by all those pixels in the texture image which have a similar neighborhood.

More generally, the best performing approaches for texture synthesis do not construct explicit generative models but rather rely on non-parametric, example-based representations (e.g. Efros and Leung, 1999; Wei and Levoy, 2000; Kwatra et al., 2003). On the one hand this prevalence of non-parametric approaches reflects the fact that suitable generative models such as MRFs are computationally rather expensive (because they require sampling to generate new textures). On the other hand, however, this is presumably also an indication of the difficulties of finding a parametric formulation that is sufficiently powerful to represent textures well.

3.4 Experiments: Comparison of the generative power of GFoE, FoE, and BiFoE

This section describes experiments that evaluate the generative power of the FoE for natural image structure on several texture modeling tasks. To gain some insight into the required properties of MRF-based texture models we first consider 1D patterns. We then compare the performance of the FoE with our extended model (BiFoE) and (as a baseline) with the much simpler GFoE on real image textures, considering a synthesis

task (section 3.2.4) as well as a texture inpainting task (section 3.4.5). We find large differences for which we give an intuitive explanation in section 3.4.6. Details of the experimental setup with respect to the dataset, the evaluation, and the parameters for learning are described in sections 3.4.1, 3.4.3, and 3.4.2 respectively.

3.4.1 Data

3.4.1.1 One-dimensional patterns

In order to gain more insight into the nature of the representations learned by the different models we investigate their ability to model a set of periodic 1D patterns, including simple sine waves, square wave patterns, a linear combination of two phase-locked sine waves of different frequencies, and a series of pulses (delta peaks). These simple patterns have the advantage that the nature of the representation learned by the model can be relatively easily understood. Examples of the patterns are shown in Fig. 3.2. The 1D patterns were typically 30 or 32 pixels wide. We varied the period of the patterns but show results only for one particular set of parameters. In all cases we added some IID Gaussian noise ($\sigma = 0.05$) to the training patterns. The patterns were chosen so as to distinguish between the simple GFoE and a more powerful model: The GFoE models the power-spectrum of the data, so it should be able to model the sine wave, but we expected it to fail, for instance, for the two phase locked sine waves, as it should be incapable of modeling the specific phase relationship between the two sinusoids. The interesting question was then whether one of the other two models would be able to account for this relationship.

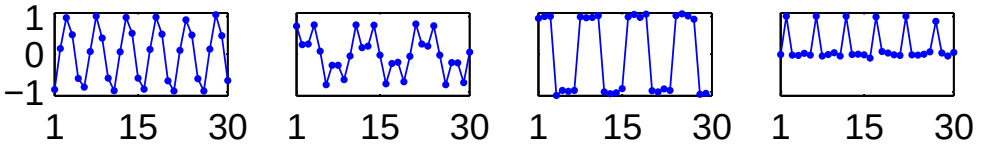


Figure 3.2: **Examples of 1D patterns**: sine wave ($f = 0.2\ cyc/pt$), phase locked sine waves ($f_0 = 0.1\ cyc/pt$, $f_1 = 0.3\ cyc/pt$), square wave (period: 8), series of pulses (pulse distance: 5pt); $\sigma_{noise} = 0.05$

3.4.1.2 Natural image textures

We use a range of six Brodatz textures (Brodatz, 1966) as well as two synthetic patterns. All textures were chosen so as to be at reasonable scale given practical filter

sizes for the FoE. Digitized versions of the Brodatz textures were downloaded from the web[2] and scaled by a factor 0.75 or 0.5 (preserving all major features of the textures). Examples of the (scaled) textures are shown in Figure 3.3 (top 6 panels). We further generated two sets of synthetic texton patterns by placing white circles and crosses randomly on a black background, ensuring that individual textons did not touch each other (the diameter of the circles was 9 pixels, the bars of the crosses were 2×10 pixels (bottom panels in Fig. 3.3). For the two synthetic texton images some IID Gaussian noise ($\sigma_{noise} = 0.05$) was added for learning.

3.4.2 Learning

In our experiments we found that it was not possible to learn BiFoE models with standard CD. To compare models on equal footing we trained both the basic FoE as well as the extended FoE using persistent chains (Tieleman, 2008). We used stochastic gradient ascent (SGA) with mini-batches, i.e. we divided the dataset into small batches, processed one batch, then updated the parameters before processing the next mini-batch. This is computationally advantageous in that it allows for more parameter updates per processed data-points.

In all cases we used 500 25×25 pixel patches, cropped randomly from the texture images of size 480×480 or 320×320 pixels (after scaling). The data was scaled to have overall mean zero (across all patches and pixels) and a standard-deviation of 1. The training data was presented in 5 mini batches containing 100 data points each.

We used 100 persistent chains (the same number as data points per mini-batch). The chains were initialized at the data points in the first mini batch and updated by one step of hybrid Monte Carlo (HMC) for each minibatch. For each step of HMC we randomly sampled a momentum from a Gaussian $N(\mathbf{0}, \boldsymbol{I})$ and then simulated the Hamiltonian dynamics for 30 leapfrog steps (cf. section 2.3.1.2). The step size was adjusted dynamically to keep the acceptance rate at 0.9.

We used a momentum of 0.9 and an initial learning rate of 0.0001. The learning rate was held fixed for the FoE. For the BiFoE we obtained better results for some textures by reducing η over the course of learning (according to a linear schedule) and for some textures it also seemed advantageous to regularly restart a fraction of the Markov chains over the course of learning.

The GFoE models were also trained with SGA (same number of mini batches as

[2] `http://www.ux.uis.no/~tranden/brodatz.html`

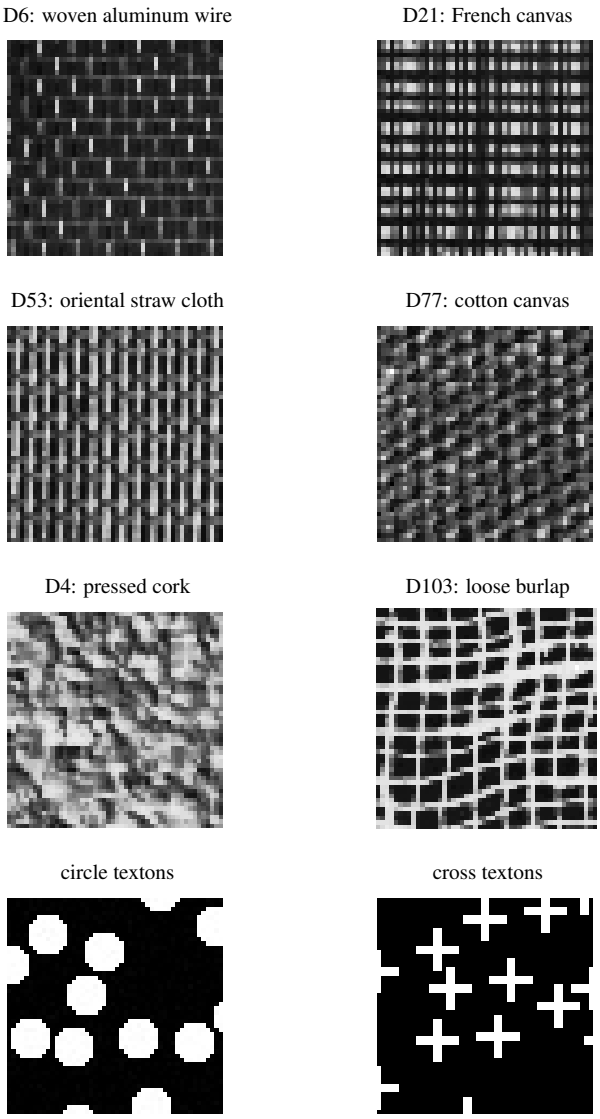


Figure 3.3: **Training data**: 50×50 patches of the image textures used in our experiments.

for the FoE models), but using the exact gradient and with a (fixed) learning rate of 0.001.

We experimented with different sizes and numbers of filters. Larger and more filters typically improved the visual quality of the models slightly (the maximum size of filters that we tried was 7×7 pixels, the maximum number was $M = 9$ or $M = 15$) but for a reasonable range of values for these two parameters we did not observe large differences in model quality. Unless noted otherwise results shown are obtained with models with $M = 9$ filters of 7×7 pixels.

As explained in Roth (2007) the patch boundaries can be problematic as the cliques centered on pixels near the boundary of a patch are incomplete (pixels of the cliques lie outside the image boundary and are effectively unobserved). Unfortunately, due to the nature of our model, it is not possible to integrate out these unobserved variables. For learning we therefore follow the approach taken by Roth (2007) and have cliques centered only on pixels sufficiently far away from the patch boundary for all cliques to be complete. This means that pixels close to the boundary will be constrained by fewer cliques and is the reason why for the learned models filter weights are typically larger near the filter boundary than in the center of the filter (see e.g. Fig. 3.11). Recently, Schmidt et al. (2010) have proposed an alternative solution to this problem in the context of standard CD: They use larger patches and perform Gibbs sampling for the negative phase conditioned on the real patch boundary (similar to our inpainting experiments described below). This approach does, however, not seem appropriate for use with *persistent* chains unless much larger images are used for the negative particles than in our experiments.

3.4.3 Evaluation

The most direct way to evaluate a FoE/GFoE/BiFoE model would be to measure the probability of a set of images under the model. However, the intractability of the partition function $Z(\Theta)$ for the FoE and BiFoE model means that this is not possible. Previous work has therefore primarily relied on "surrogate tasks" such as denoising or infilling of small image regions. These are, however, relatively indirect measures of the quality of a generative model. A more direct approach is to draw samples from the learned models and compare them to the training data. This approach was pursued e.g. by Zhu et al. (1998) and Zhu and Mumford (1997), and has more recently also been adopted by Schmidt et al. (2010); Ranzato et al. (2010b). Our main method of

evaluation is thus to draw samples from the models and compare these samples with the original textures. In section 3.4.5 we also discuss texture inpainting experiments (unlike in most work on image priors, however, we require the models to fill in large regions, e.g. of size 50×50 pixels).

Samples were generated by initializing Markov chains with IID Gaussian noise $N(0, \boldsymbol{I})$, and performing MCMC (with HMC as during learning) until the chains had settled to equilibrium. For the GFoE samples were drawn from the corresponding Gaussian distribution directly.

For a quantitative assessment of model quality we used a texture similarity score (TSS) based on normalized cross correlation (NCC). Specifically we sampled 100 texture patches of size 25×25 pixels from the models. In order to reduce the influence of boundary pixels we discarded 3 pixels on all four sides, resulting in patches of size 19×19 pixels. We then computed the NCC with the original texture image and used the maximum value across the image as the similarity score for the respective texture sample:

$$S(\mathbf{x}, \mathbf{s}) = \max_i \frac{\mathbf{x}_{(i)}^T \mathbf{s}}{\| \mathbf{x}_{(i)} \| \| \mathbf{s} \|} \tag{3.18}$$

where $\| \cdot \|$ is the L_2-Norm, $\mathbf{s}$ is the texture sample drawn from the model and $\mathbf{x}$ is the original texture image of size 320×320 or 480×480 pixels.

The TSS provides an indication of how similar the samples from the model are to the training data. Thus, a high TSS suggests that the model produces predominantly samples that are looking like the real texture. This is, however, not enough, since the model could produce only a single sample which happens to be close to one particular texture patch, but ignore the full variability of the true texture. For the BiFoE we therefore also perform the reverse test, which is meant to give an indication of how well the model captures the variability of the real texture. For this purpose we randomly select 500 patches from the real texture. Each of these patches we then compare with 500 patches extracted from samples from the model (again, using NCC) and determine the most similar sample patch for each real texture patch. The corresponding score reflects how well the real texture patches are matched by samples from the model. Furthermore, for each real texture patch we also extract another 500 patches from the original texture image (such that they do not overlap with the patch under consideration) and compute the TSS for those as well. This second score provides us with a baseline (calibration) for the comparison between the model and the real texture: It is the score we would expect if the model produced samples equivalent to the real texture.

TSSs were computed only for the first four textures in Fig. 3.3a (D6, D21, D53, and D77), as these four textures were sufficiently regular for the correlation-based similarity score to be meaningful. For these textures we found that the TSS is generally in reasonably good agreement with the visual quality of a sample as is demonstrated in Fig. A.1 in Appendix A.3.

3.4.4 Experiment 1: Texture synthesis

3.4.4.1 Results for one-dimensional patterns

As a first test of the BiFoE model we applied it to the set of relatively simple 1D patterns described in section 3.4.1.1 above. As pointed out before, these patterns were chosen such as to highlight the differences between the models: We expected that the GFoE would be able to model the sine-wave pattern (which is fully described by its power-spectrum) but none of the other, more complicated patterns which require modeling the relative phases of different frequency components. The simple patterns further have the advantage that they allow us to inspect and understand the nature of the representations learned.

All models were trained as described in section 3.4.2 although typically fewer epochs were required. GFoE models were trained by SGA on the true likelihood gradient. Unless otherwise noted we show results for models with $M = 9$ 7-point filters.

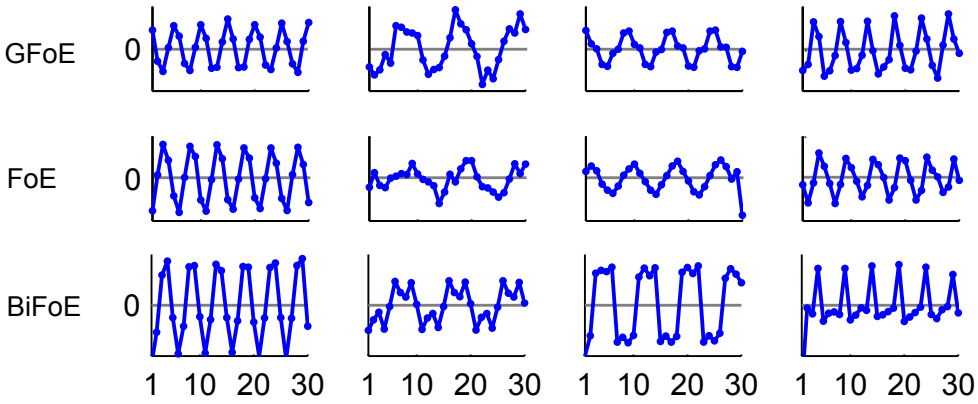


Figure 3.4: Samples drawn from GFoE (*top*), FoE (*middle*) and BiFoE (*bottom*) models trained on the 1D patterns shown in Fig. 3.2. Left to right: sine wave, phase locked sine waves, square wave, pulses.

GFoE / FoE: Fig. 3.4 shows representative examples of the patterns generated by the (G)FoE models trained on the four 1D patterns. It is clear that both models fail

except for the sine wave. For the GFoE this is what we expected and the learned representation can be easily verified by computing the principal components of the covariance matrix: For the sine wave, for instance, there are two relevant components, corresponding to a sine/cosine pair at the appropriate frequency; for the phase locked sinusoids there are four such components corresponding to two sine / cosine pairs at the frequencies of the component sinusoids (cf. Fig. 3.5). Yet, as expected and confirmed by the samples, even though the model does recover the relevant frequency components, it does not account for their specific phase relationship and thus fails to reproduce the phase-locked pattern accurately. Comparing the sets of samples generated from the FoE and the GFoE there is no obvious difference, suggesting that the FoE does not provide any advantage when modeling simple patterns as the ones above (note that Fig. 3.4 shows only one sample per model and pattern).

BiFoE: BiFoE models were reliably learned for all four 1D patterns, as is shown in Fig. 3.4, and the parameters were often interpretable. This is illustrated in Fig. 3.6 which shows the experts (filters and potential functions) for a model of the square wave pattern. It is easy to see how the filters together with the corresponding potentials, one of them being bimodal, constrain the patterns generated by this model. While it is easy to see how the square wave can be modeled using filters as the ones shown in the figure and *bimodal* potentials, it is not obvious how appropriate filters could be constructed when only unimodal, zero-centered potentials are allowed. Note that in order to obtain more easily interpretable results this model was trained with the minimum number of filters for the square wave, and with periodic boundary conditions. Details are provided in the figure legend.

3.4.4.2 Results for image textures

The experiments with one-dimensional patterns confirmed our expectations that the GFoE would lack the flexibility to model any of the more complicated patterns. More interestingly, we found that the FoE did not seem to hold any advantage over the GFoE for this problem: In terms of the visual quality of the samples the results for the FoE were no better than for the GFoE. At the same time, the BiFoE, even though it adds only two parameters per expert, models all patterns without difficulties.

Obviously, we are not interested in modeling such stereotypical one-dimensional patterns but real image textures. The question was thus whether the advantages of the BiFoE would carry over to this more complex task. We trained models on the image textures shown in Fig. 3.3 as described in section 3.4.2. We then evaluated the learned

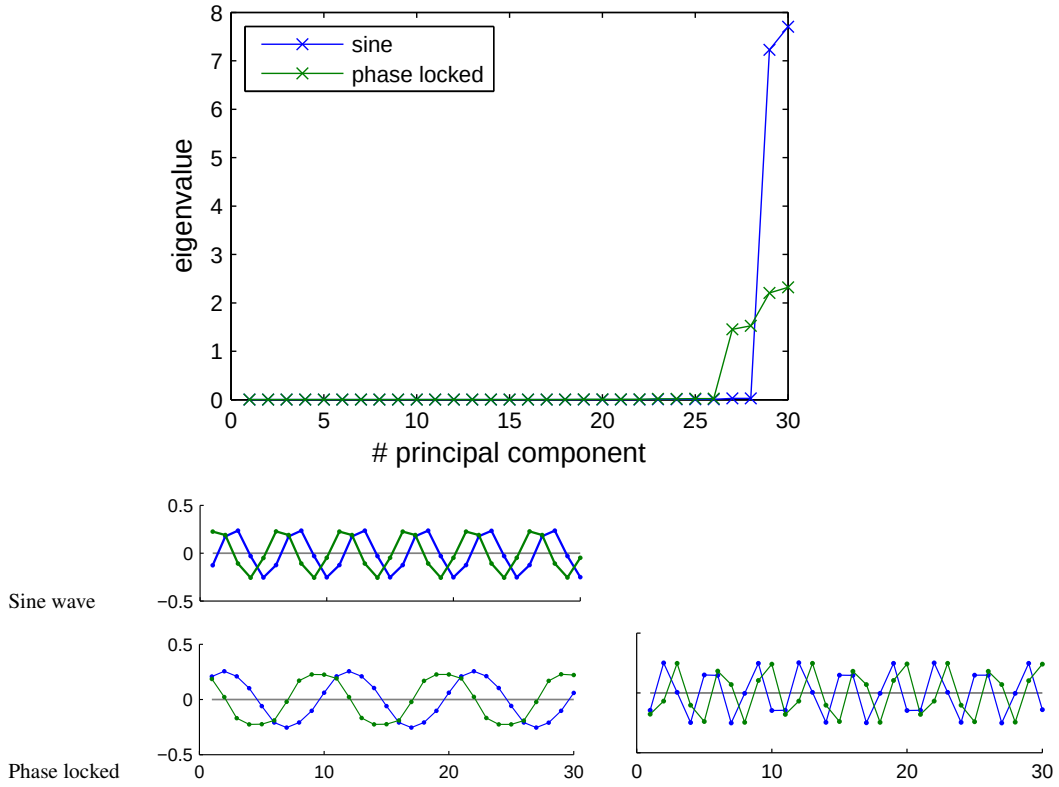


Figure 3.5: Illustration of the representations learned by the GFoE models for the sine-wave and the phase-locked patterns. *Top*: Eigenvalue spectrum of the covariance matrix computed from the learned filters. *Bottom*: Significant prinicpal components for the sine-wave pattern and the phase-locked pattern respectively. For the sine-wave pattern there are only two significant components (a sine/cosine-pair of the appropriate frequency). For the phase-locked pattern the covariance matrix has four significant components (two sine/cosine-pairs), the two most significant are shown on in the plot on the left, the 3rd and 4th are shown on the right.

models by drawing samples and computed the TSS described in section 3.4.3. For visualization we further sampled 50×50 pixel patches and removed 2 pixels on all sides.

Fig. 3.9a shows boxplots of correlation scores for the three models for the first four textures in Fig. 3.3 (D6, D21, D53, D77). Samples from the models are shown in Figures 3.7 and 3.8 for the (G)FoE models and the BiFoE models respectively. From the low correlation scores and from comparing samples in Figures 3.7, 3.8 with the originals in Fig. 3.3 it is clear that GFoE and FoE models fail equally to reproduce three of the four textures. For D77 the GFoE / FoE models were able to produce

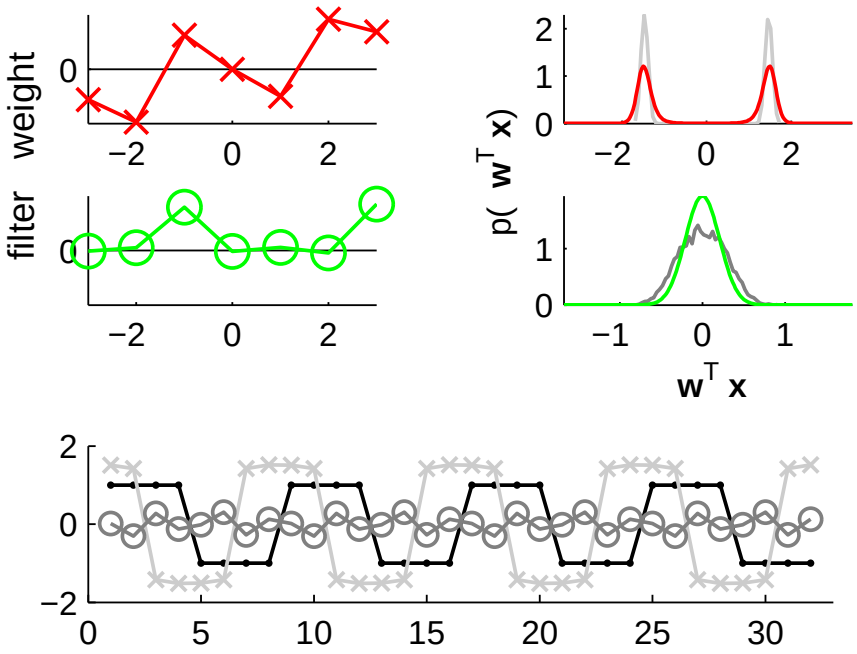


Figure 3.6: **BiFoE model learned for a square wave pattern with period 8**. $M = 2$ experts, filter width 7, trained with periodic boundary conditions. *Top left*: Filters $\mathbf{w}_1$, $\mathbf{w}_2$ of the two experts (shown in red/crosses and green/circles). *Top right*: Potential functions of the experts (also shown in red and green respectively) and corresponding response histograms of the two filters (light and dark gray). *Bottom*: The first filter responds with either -1.5 or 1.5 as it is shifted along the square wave. The second filter responds with (approximately) zero everywhere. Square wave: black; response of first filter: light gray/crosses; response of second filter: dark gray/circles. In an idealized (noise free) form the two filters are $\mathbf{w}_1 = (\ -c\ \ -d\ \ c\ \ 0\ \ -c\ \ d\ \ c\ )^T$ and $\mathbf{w}_2 = (\ 0\ \ 0\ \ e\ \ 0\ \ 0\ \ 0\ \ e\ )^T$ where $c, d, e \neq 0$ can be chosen arbitrarily. The two experts together constrain the patterns generated by the model to be (noisy) square waves.

reasonable samples, although not very consistently.

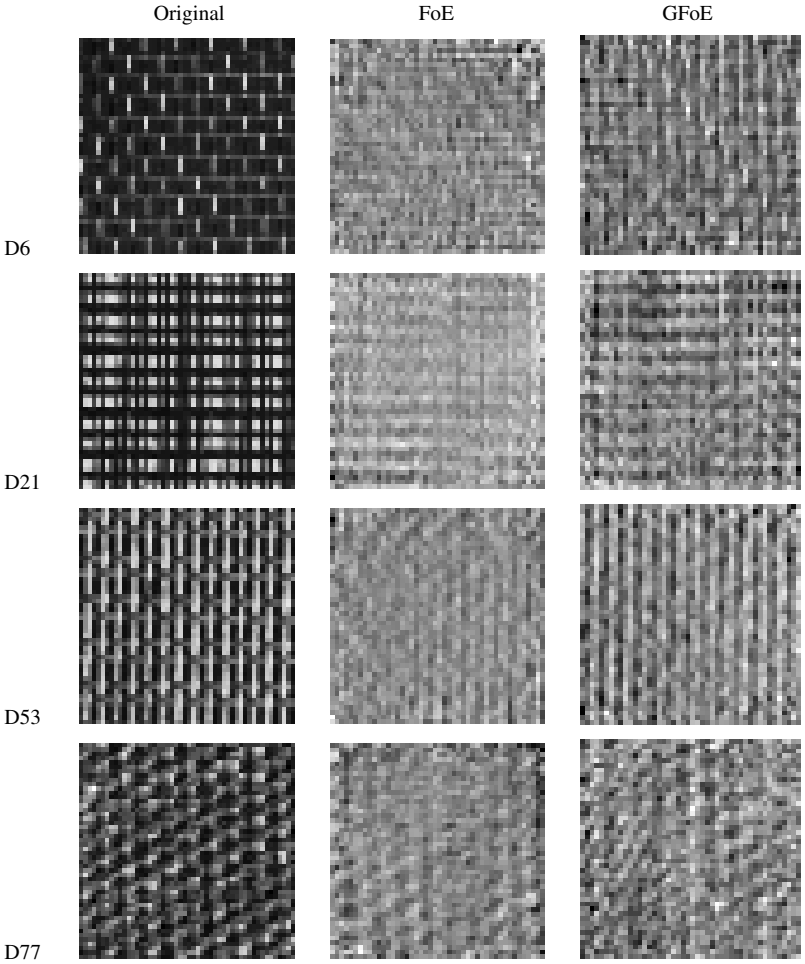


Figure 3.7: FoE samples (*center*) and GFoE samples (*right*): 46×46 samples for the FoE and GFoE models trained on textures D6, D21, D53, and D77 shown in Figure 3.3 (to faciliate the comparision examples of the original textures have been reproduced on the *left*). Neither the FoE nor the GFoE samples are good representatives of the original textures.

While the FoE is not performing any better than the GFoE model, results for the BiFoE are rather different: For the first three textures the TSSs are clearly higher than for the GFoE / FoE. The difference for D77 is smaller, but the quality of the samples from the BiFoE is much more consistent than for the GFoE or FoE models and visually the samples are much more convincing. Fig. 3.8 shows representative 50×50 samples drawn from the BiFoE model distributions for all 6 Brodatz textures described above. Especially for the first 5 textures it is difficult to distinguish samples drawn from the

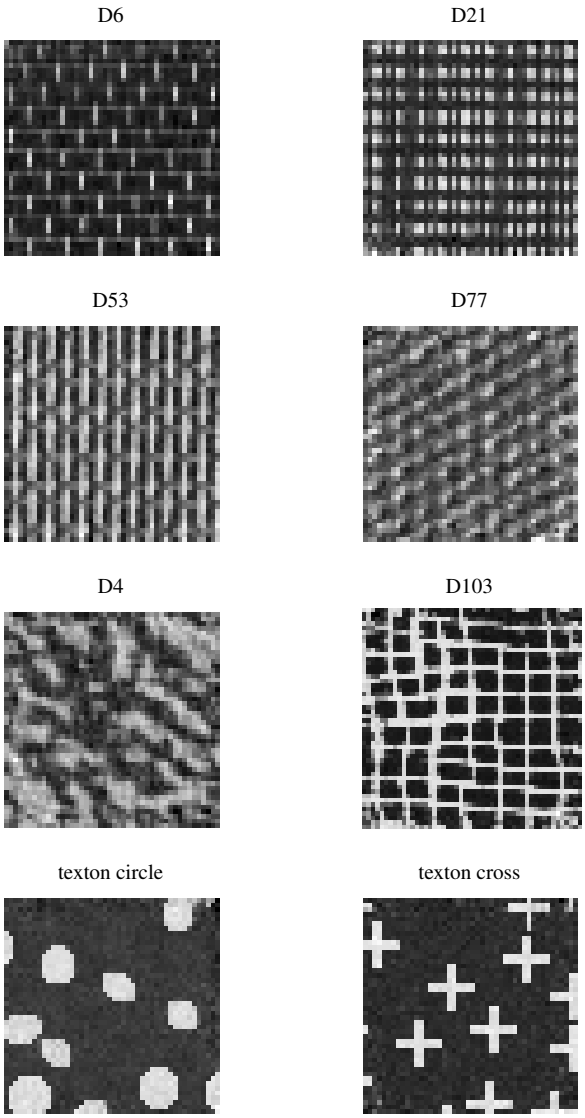
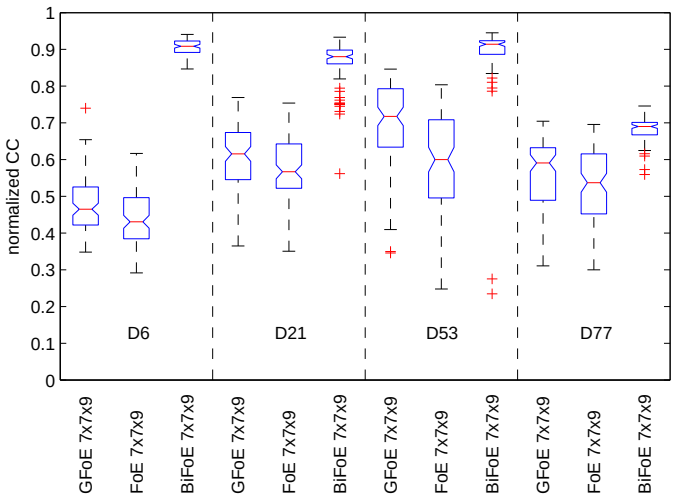
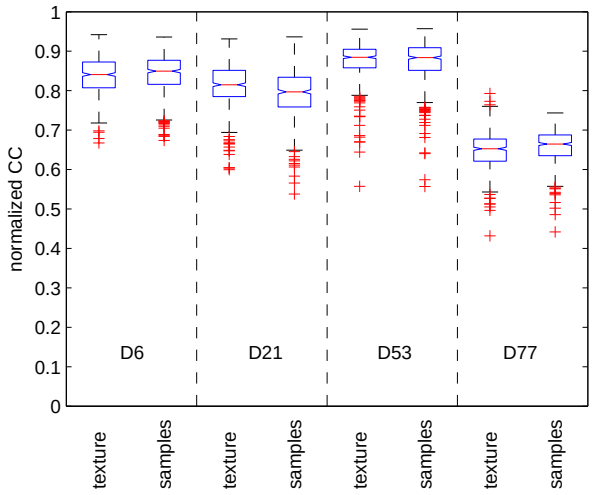


Figure 3.8: BiFoE samples: samples drawn from BiFoE models trained on the textures in Fig. 3.3.



(a)



(b)

Figure 3.9: *(a)* TSSs for the four textures D6, D21, D53, and D77 for the three models considered. The three columns for each texture correspond to (from left to right) the GFoE, FoE, and BiFoE respectively. Boxes indicate the upper and lower quartiles as well as the median (red bar) of the TSS distributions; whiskers show extent of the rest of the data; red crosses: outliers. (100 data points in each case). *(b)* Similarity scores for matching patches from the original texture to other non-overlapping patches from the original texture (left box in each group) and to samples from the learned BiFoE models (right box in each group). Samples from the BiFoE models match the original texture patches as well as other parts of the original texture do, suggesting that the models indeed largely explain the variability present in the training data.

model from the original textures. The two texton patterns are also modeled well (although the sample quality is somewhat less consistent). For the texton models shown in Figure 3.3 (bottom panels) we used $M = 15$ 7×7 filters as this gave better results. The BiFoE is able to model not only the relatively regular textures (D6, D21, D53, D77) but can also handle more variable ones.

The fact that we can draw samples that are larger (twice as large in the Figures shown here, but potentially of arbitrary size) than the patches used for training confirms that the model is not just memorizing the training data. To further corroborate that the model covers the full variability of the training data we also computed the "reverse TSS" explained in section 3.4.3: For a given set of patches from the real texture we determined how well these were matched by samples from the model. The results are shown in Figure 3.9b and suggest that for all four textures for which we computed scores we can find for each real texture patch a sample that matches this real texture patch as well as the best matching (but non-overlapping) patch extracted from the real texture. This suggests that the models not only produce samples that look like patches from the original textures, but that they also cover the variability of the original textures reasonably well.

3.4.5 Experiment 2: Constrained Texture Synthesis

One interesting problem that a generative model can be applied to is constrained texture synthesis such as filling in a hole in a texture image. This requires synthesizing texture for those parts of the image that are missing in a way that is consistent with given parts of the image. Inpainting is an appealing task for evaluating a model since it imposes constraints on the stochasticity, but it also has many practical applications such as removing an object shown in front of a textured background. For our experiments we used 70×70 texture images with a 50×50 hole in the center. In addition to the probabilistic models discussed in the previous section we also included the synthesis method proposed by Efros and Leung (1999) in this experiment in order to compare the BiFoE to a state-of-the-art non-parametric approach[3].

For the GFoE, FoE, and BiFoE we sampled the "missing" pixels conditioned on the "existing" pixels with HMC-MCMC. The missing pixels were initialized to 0 for all textures (except for the circle textons for the BiFoE: in this case we initialized the

[3]We only compared with Efros & Leung's method on the inpainting task because this method requires a "seed" for the synthesized texture which is naturally given by the inpainting frame in the inpainting task.

missing pixels with IID Gaussian noise). Our implementation of Efros & Leung's (E&L) method used 15×15 pixel patches (referred to as "neighborhood windows" in Efros and Leung, 1999) for infilling extracted from those image patches used to train the BiFoE models; we experimented with different patch sizes for E&L's method and a size of 15 seemed to give good results. We used 20 different texture images for each texture, and repeated inpainting 5 times for each image since all methods are stochastic. The quality of the results across repetitions for a given texture image was typically very consistent. We performed inpainting for the four regular textures and computed the NCC between the original texture image and the inpainting result (cf. section 3.4.3). Results are shown in Fig. 3.10. The NCC values (averaged over repetitions and images for each texture and method) for the four regular textures are given in the table below (NCC $\pm$ std-dev)[4]:

	GFoE	FoE	BiFoE	Efros & Leung
D6	0.7245 ± 0.0261	0.6686 ± 0.0385	0.8769 ± 0.0163	0.8300 ± 0.0380
D21	0.7862 ± 0.0237	0.7971 ± 0.0283	0.8653 ± 0.0244	0.8330 ± 0.0351
D53	0.7736 ± 0.0208	0.7808 ± 0.0159	0.9145 ± 0.0125	0.8878 ± 0.0300
D77	0.5675 ± 0.0286	0.6102 ± 0.0229	0.6567 ± 0.0205	0.6325 ± 0.0490

Two observations can be made: Firstly, the GFoE and FoE perform better than would be predicted from the results in the previous section. If provided with a reference (the inpainting frame), they can "maintain" the corresponding structure over a certain distance, although the quality of the texture decreases as the distance to the closest reference pixels increases. This can be seen in Figure 3.10: the textures generated by the GFoE and FoE models are quite blurry in the center of the image. For the GFoE this behavior is expected: It models the power spectrum of an image while ignoring the phases, but in the inpainting case the phases are to some extent imposed upon the generated texture by the inpainting frame. Secondly, BiFoE and E&L's method both perform very well on almost all textures – and considerably better than the GFoE / FoE. They do not suffer from the degradation of the texture toward the center of the completed image. The BiFoE seems to perform even slightly better than E&L's method, but more importantly, in contrast to the latter the BiFoE is formulated as an explicit parametric generative model that absorbs the characteristics of a texture into a compact representation (only 9 filters of size 7×7 plus 9×3 parameters for the experts' potentials).

[4]Using the root mean squared error between the samples and the original image instead of the NCC leads to very similar results in terms of the relative quality of the different models .

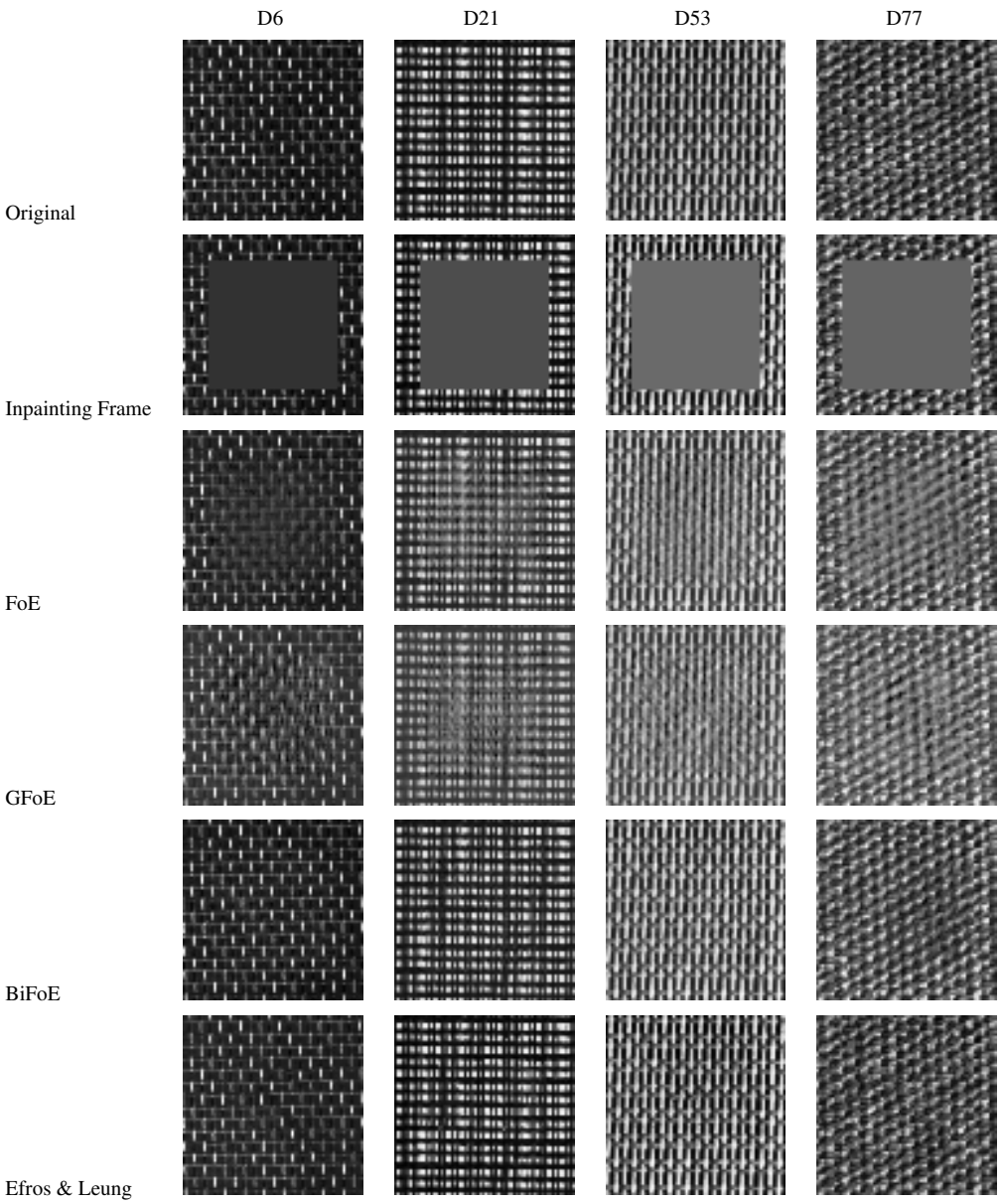


Figure 3.10: Inpainting results for all regular textures

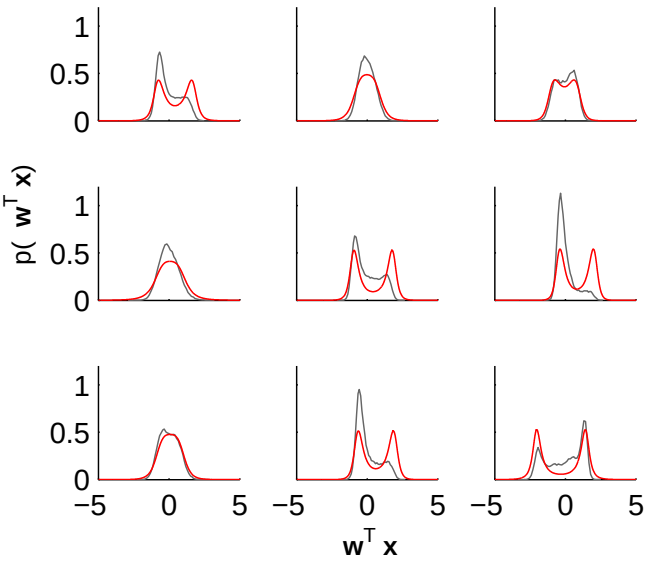
Top to bottom: Original image; inpainting frame; image completed by FoE; image completed with GFoE; image completed BiFoE; image completed by Efros & Leung's method

3.4.6 Understanding the differences between the models

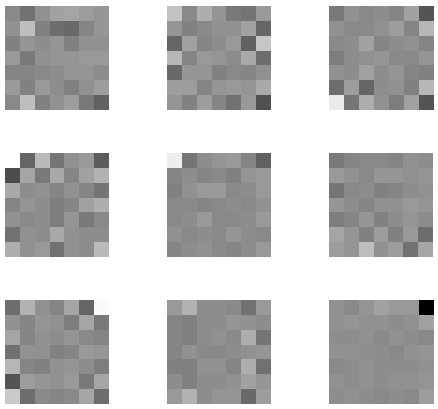
From the results presented above it is clear that the BiFoE learns texture models that are markedly superior to the ones learned by the FoE and the GFoE, but there seems to be little difference between the FoE and the GFoE.

Some insight into why this is the case can be gained from considering the models learned for simple 1D patterns. While all three models can describe the 1D sinusoid (a sinusoidal pattern with random phase is fully described by its power spectrum and we would therefore expect it to be modeled well even by the simplest model, the GFoE), only the BiFoE is able to model any of the more complex patterns. Figure 3.6 shows the minimal BiFoE model that can describe the square wave pattern – the learned representation is heavily reliant on the bimodality of the potential functions learned by the BiFoE, and this bimodality cannot be represented with the GFoE or the simple FoE. Inspecting the BiFoE parameters learned for real textures we find that the additional flexibility provided by the bimodal expert function is indeed being exploited: All of the models learn several experts with bimodal expert functions (i.e. $a_j < 0$). The interactions between the learned bimodal experts give rise to heavily skewed and in many cases also bimodal response marginals (because of the interactions the response histograms typically deviate from the learned potential functions). This is illustrated in Fig. 3.11 for the model learned for texture D53 (similar results for the other models can be found in the appendix, section A.4). The learned BiFoE models are thus very different from the FoE (and GFoE) models for which the response marginals are almost exclusively centered at zero and roughly symmetric.

The implications of these findings in terms of the resulting probability distributions over images are best understood by considering the possible interactions between the experts and the different types of potential functions. For the GFoE the probability density function arising from the replicated experts will always be Gaussian (cf. section 3.4.3) and is thus inherently unimodal. For the FoE in the form of equation (3.10) this probability density function can take more interesting forms, however, because all expert potentials are centered at zero, the overall probability density function will still always be unimodal independent of the number of experts and the parameters they learn. It can be shown that the energy function always has a unique minimum at $\mathbf{x} = \mathbf{0}$ (as long as the overall model is complete; proof in Appendix A.1). The potentials of the BiFoE, in contrast, allow for much more flexibility in shaping a potentially multimodal pdf. This idea is illustrated in Figure 3.12 for the non-translation invariant



(a)



(b)

Figure 3.11: Parameters of the BiFoE model for texture D53. *Top*: Expert nonlinearities for the nine experts used (*red*) and filter response marginals of the corresponding filters for the training data (*light gray*). *Bottom*: Corresponding filters.

(PoE) case (see also Hinton, 2002 for a similar example). The translation-invariant FoE is effectively a product of $N \times M$ experts (where N is the number of image pixels and M the number of different experts) so the same considerations hold in principle, but it is necessary to take the interactions between the replicated versions of the M experts into account explicitly.

To make the above intuition more concrete it is helpful to consider a model with a slightly different form of the bimodal potentials which is more amenable to an ana-

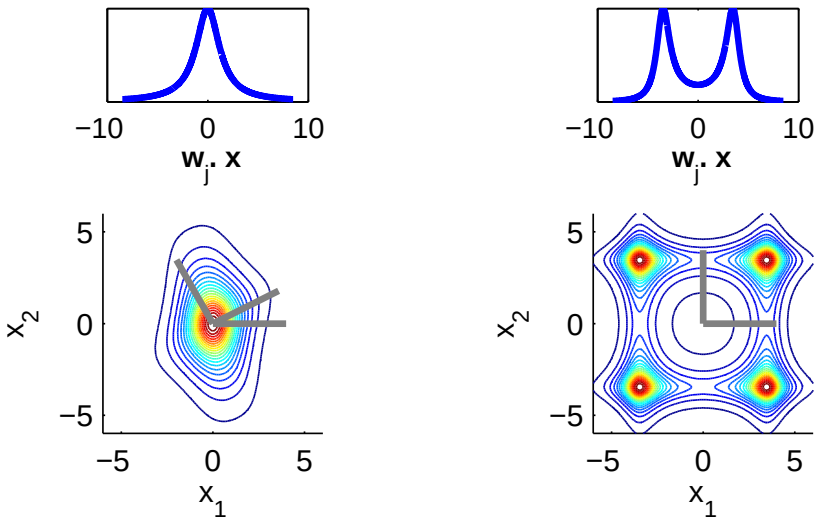


Figure 3.12: (a) Contour plot of the pdf of an (overcomplete) PoE model in $\mathbb{R}^2$ obtained from three Student-t experts with zero centered potentials (the expert function is shown on the left; $\mathbf{w}_{1\ldots3}$ are superimposed on the contour in dark gray). (b) Same as in (a) but for two bimodal experts. The pdf in (a) is unimodal, the one in (b) multimodal.

lytical investigation: In this model, which we will refer to as the mixture of Gaussians (MoG)-BiFoE below, the expert function is a mixture of two one-dimensional Gaussians of equal variance with means that are 2Δ apart

$$\Phi(y) = \frac{1}{2}\exp\left\{-\frac{1}{2}(y-b+\Delta)^2\right\} + \frac{1}{2}\exp\left\{-\frac{1}{2}(y-b-\Delta)^2\right\}. \tag{3.19}$$

The b and the Δ in this expression play a role similar to the b and the a in the bimodal expert function described above (eq. 3.16) in that they control the overall position and the separation of the two modes at $y = b \pm \Delta$ respectively.

The energy of an image is thus given by

$$E_{\mathrm{MoG}}(\mathbf{x}) = -\sum_i \sum_j \log\left\{\exp\left[-\frac{1}{2}\left(\mathbf{w}_j^T\mathbf{x}_{(i)} - b_j + \Delta_j\right)^2\right] + \exp\left[-\frac{1}{2}\left(\mathbf{w}_j^t\mathbf{x}_{(i)} - b_j - \Delta_j\right)^2\right]\right\}. \tag{3.20}$$

This energy can be written in terms of auxiliary latent variables $z_{ij} \in \{0,1\}$ that select, for expert j and image pixel i, which of the two modes is "active":

$$E^{\mathrm{Aux}}_{\mathrm{MoG}}(\mathbf{x},\mathbf{z}) = \sum_{i,j}\left[\frac{z_{ij}}{2}(\mathbf{w}_j^T\mathbf{x}_{(i)} - b_j + \Delta_j)^2 + \frac{(1-z_{ij})}{2}(\mathbf{w}_j^T\mathbf{x}_{(i)} - b_j - \Delta_j)^2\right]. \tag{3.21}$$

E_{MoG} is obtained as the free energy of $\mathbf{x}$ in the joint distribution $p(\mathbf{x},\mathbf{z}) = \frac{1}{Z}\exp\{-E^{\text{Aux}}_{\text{MoG}}(\mathbf{x},\mathbf{z}))\}$ by summing out the latent variables $\mathbf{z}$ (cf. section A.2 in the Appendix).

This formulation gives rise to tractable conditional distributions: The z_{ij} are conditionally independent given an image $\mathbf{x}$ (cf. Appendix A.2). Furthermore, given a state of all z_{ij}s the distribution over the image is a Gaussian distribution $\mathbf{x}|\mathbf{z} \sim N(\boldsymbol{\mu}(\mathbf{z}),\Sigma)$ with mean

$$\boldsymbol{\mu}(\mathbf{z}) = \left(\sum_j W_j^T W_j\right)^{-1} \sum_j W_j^T \mathbf{1}(b_j + \Delta_j) - 2\left(\sum_j W_j^T W_j\right)^{-1} \sum_j W_j^T \mathbf{z}_j \Delta_j \qquad (3.22)$$

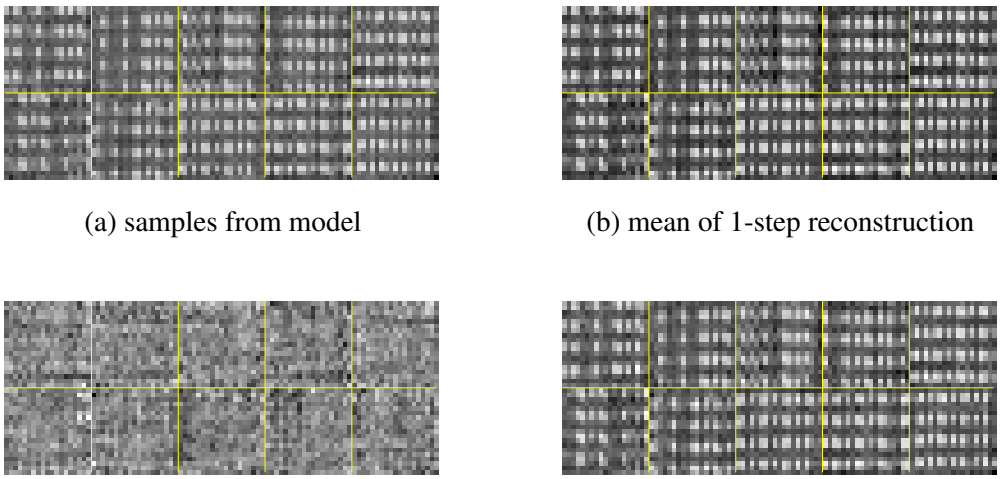
and covariance

$$\Sigma = \left(\sum_j W_j^T W_j\right)^{-1}. \qquad (3.23)$$

Thus, the overall model is a mixture of Gaussians with an exponential number of components (one component for each configuration of the latent variables) and these components all have the same covariance but different means, consistent with the multimodal representation suggested above.

To assess this idea in practice we trained a MoG-BiFoE model on some of the textures used in the experiments above. Overall, the MoG-BiFoE produced results that were visually worse than the ones obtained with the BiFoE model. In some cases, however, the results were sufficiently good to perform an analysis of the nature of the representation learned, in particular of the role of individual modes and of the relative contributions of the mean and the covariance in generating the observed structure. Below we will show results for texture D21. Since the z_{ij} are not marginally independent we obtained samples of the z_{ij} by first sampling texture patches from the trained MoG-BiFoE, using the formulation in eq. 3.20 and using HMC as for the standard BiFoE. For each texture patch generated in this way we then obtained a sample of the the latent variables z_{ij} using the conditional distribution given in the Appendix (eq. A.11). Subsequently we computed the conditional mean, and the conditional covariance using equations (3.22) and (3.23). The results are shown in Fig. 3.13. The figure suggests that most structure present in the sample $\mathbf{x} \sim \mathbf{x}|\mathbf{z}$ is present already in the mean of the conditional distribution. One way to quantify and compare the relative contribution of the mean and of the random component arising from $N(\mathbf{0},\Sigma)$ is by considering $\boldsymbol{\mu}^T\boldsymbol{\mu}$ and $\text{trace}(\Sigma) = E[\mathbf{y}^T\mathbf{y}]$ where $\mathbf{y} \sim N(\mathbf{0},\Sigma)$. For all 10 samples considered in Fig. 3.13 $\boldsymbol{\mu}^T\boldsymbol{\mu} > 550$ whereas $\text{trace}(\Sigma) = 72$. Overall these results suggest that, at least in this particular case, different instances of the texture arise primarily from choosing different components of the global mixture learned by the model. As pointed out above,

the components differ with respect to their means but share the same covariance matrix and the contribution of the covariance to the structure seen in the samples is small. The effective number of distinct mixture components is hard to determine, but it should be noted that the potential number of mixture components is exponential in the number of latent variables and thus also grows exponentially with the size of the image. The standard BiFoE proposed in section 3.2.2.3 has a different parametric form and the analysis does not carry over directly, nevertheless these results are likely to provide at least a partial explanation of how the model might be operating.



(a) samples from model

(b) mean of 1-step reconstruction

(c) random component of 1-step reconstruction

(d) full 1-step reconstruction

Figure 3.13: **Analysis of the MoG-BiFoE model for texture D21**. *(a)* 10 samples from a MoG-BiFoE model (energy as given in equation 3.20) trained on texture D21 (same training procedure as for standard BiFoE). The samples were obtained using HMC. For this set of texture samples the one-step reconstruction distribution was obtained by first sampling the latent states conditioned on the texture patches and then computing the conditional distribution over the visibles according to equations (3.22) and (3.23). *(b)* shows the conditional mean $\boldsymbol{\mu}(\mathbf{z})$; *(c)* illustrates the nature of the covariance matrix by showing one sample from the zero mean-Gaussian $N(\mathbf{0},\Sigma)$; *(d)* shows a full sample from the one-step reconstruction distribution $N(\boldsymbol{\mu}(\mathbf{z}),\Sigma)$.

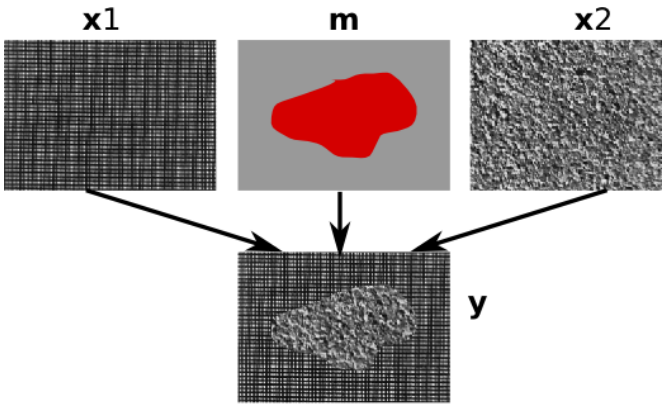


Figure 3.14: **Illustration of the hierarchical model with two texture regions**. The visible image $\mathbf{y}$ is obtained by combining the two homogeneous texture images $\mathbf{x}_{1,2}$ according to the state of the mask variables $\mathbf{m}$.

3.5 Hierarchical, region-based BiFoE

In this section we will demonstrate how the BiFoE can be used a building block of a more comprehensive, hierarchical model that accounts for images with multiple texture regions. This is achieved by associating a selection variable – a mask as e.g. in Jojic and Frey (2001) – with each pixel that determines from which texture this pixel has been drawn. Thus, in the simplest case of only two image regions, there will be two latent images, one latent image for each texture and each governed by a single texture model; the observed image is a composition of these two latent images and each pixel in the observed image is chosen from one of the latent images as determined by the mask. This idea is illustrated in Fig. 3.14.

3.5.1 Model

Denoting the observed image with $\mathbf{y} \in \mathbb{R}^N$, the (binary) mask with $\mathbf{m} \in \{0,1\}^N$ and the two texture models governing the latent images $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^N$as $p_1(\mathbf{x}_1)$ and $p_2(\mathbf{x}_2)$ the full regions-model is given as:

$$p(\mathbf{y}) = \sum_{\mathbf{m}} \int d\mathbf{x}_1 \int d\mathbf{x}_2 p_M(\mathbf{m}) p_1(\mathbf{x}_1) p_2(\mathbf{x}_2) \prod_{i=1}^{N} \delta(y_i, x_{1,i})^{m_i} \delta(y_i, x_{2,i})^{1-m_i}, \qquad (3.24)$$

where $p_M(\mathbf{m})$ defines a prior over the mask, e.g. a MRF. $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ are set to be two BiFoE models trained on different textures, i.e. $p_k(\mathbf{x}) = p_{Bi}(\mathbf{x}; \Theta_k)$. This formulation might seem a bit daunting since it effectively involves a pixel-wise mixture

over distributions for which we cannot compute the normalization constants, but as shown in the Appendix (section A.5) Gibbs sampling can still be used to perform inference (which involves inferring the state of the mask and of the two latent images given an observed image $\mathbf{y}$), updating all three variables associated with pixel i (m_i, $x_{1,i}$, and $x_{2,i}$) simultaneously in each step:

$$p(m_i, x_{1i}, x_{2i}|\mathbf{y}, \mathbf{m}_{\setminus i}, \mathbf{x}_{1\setminus i}, \mathbf{x}_{2\setminus i}) = p(m_i|\mathbf{y}, \mathbf{m}_{\setminus i}, \mathbf{x}_{1\setminus i}, \mathbf{x}_{2\setminus i}) p(x_{1i}, x_{2i}|\mathbf{y}, m_i, \mathbf{m}_{\setminus i}, \mathbf{x}_{1\setminus i}, \mathbf{x}_{2\setminus i}) \tag{3.25}$$

where

$$p(m_i = 1|\mathbf{y}, \mathbf{m}_{\setminus i}, \mathbf{x}_{1\setminus i}, \mathbf{x}_{2\setminus i}) = \frac{p_M(m_i = 1|\mathbf{m}_{\setminus i}) p_1(y_i|\mathbf{x}_{1\setminus i})}{p_M(m_i = 1|\mathbf{m}_{\setminus i}) p_1(y_i|\mathbf{x}_{1\setminus i}) + p_M(m_i = 0|\mathbf{m}_{\setminus i}) p_2(y_i|\mathbf{x}_{2\setminus i})} \tag{3.26}$$

$$p(x_{1i}, x_{2i}|\mathbf{y}, \mathbf{m}, \mathbf{x}_{1\setminus i}, \mathbf{x}_{2\setminus i}) = \begin{cases} \delta(x_{1i} = y_i) p_2(x_{2i}|\mathbf{x}_{2\setminus i}) & \text{if } m_i = 1 \\ \delta(x_{2i} = y_i) p_1(x_{1i}|\mathbf{x}_{\setminus i}) & \text{if } m_i = 0, \end{cases} \tag{3.27}$$

where $\mathbf{x}_{\setminus i}$ denotes all elements of $\mathbf{x}$ except for the i-th element, i.e. $x_1, \ldots x_{i-1}, x_{i+1}, \ldots x_N$ (and similarly for $\mathbf{m}_{\setminus i}$). The expressions (3.26) and (3.27) still involve the conditional distributions $p_j(x_i|\mathbf{x}_{\setminus i})$ which might not be analytically tractable and which cannot necessarily be sampled from directly (this problem arises for instance in the case of the BiFoE). However, since these are one-dimensional distributions they can be approximated e.g. by discretization of the range of gray values (a strategy that is, for instance, also used by Roth, 2007 to sample from the standard FoE).

One important property of the formulation in equation 3.24 is that it avoids problems that would typically appear at region boundaries where neither of the two texture models would explain the data well. As the two texture models act on the two homogeneous latent images such problems do not arise. It should be noted that the formulation is very general and allows for more interesting mask priors (e.g. incorporating some knowledge about object shape) to be used. Also, although only two regions are considered in 3.24, conceptually the model is easily extended to deal with more than two. Finally, the texture models p_1, p_2 can be replaced by more powerful models, e.g. a latent variable formulation of the BiFoE in which the effective parameters Θ are determined by the state of the latent variables (see also section 3.6 below).

"Double MRFs" such as the hierarchical BiFoE have been considered for texture segmentation by several authors, e.g. Derin and Elliott (1987), Manjunath et al. (1990), Zhang et al. (1994), and Melas and Wilson (2002). In all cases the focus is, however, on solving the segmentation problem and the texture models used are consid-

erably simpler (notably Gaussian MRFs in Derin and Elliott, 1987, Manjunath et al., 1990, and Melas and Wilson, 2002). The emphasis is put on approximations that allow for efficient inference which seem to involve relatively strong simplifications in some cases: Manjunath et al. (1990), for instance, evaluate the problematic likelihood term by densely covering the image with small overlapping rectangular windows and then evaluate the GMRF likelihoods separately for each such window assuming that all pixels in that window belong to the same texture and making toroidal boundary assumptions (the normalization constant can be evaluated for each such window). The inference scheme described above for the hierarchical BiFoE is computationally potentially more involved, but conceptually much simpler and can be applied to the non-Gaussian case. Zhang et al. (1994) and Melas and Wilson (2002) also consider the possibility of unsupervised learning in such models. More generally, the idea of using a mask to combine multiple models has a long history in computer vision, e.g. in the context of layered image models which will be discussed in more detail in the related work sections of chapters 4 and 5. In those chapters we will present the *Masked RBM*, which uses a similar formulation for non-translation invariant RBMs and, in particular, we will show how a powerful model of region *shape* can be obtained in this context.

3.5.2 Experiments

In order to demonstrate the viability of the formulation in section 3.5.1 we investigated its performance on a simple (supervised) texture segmentation task.

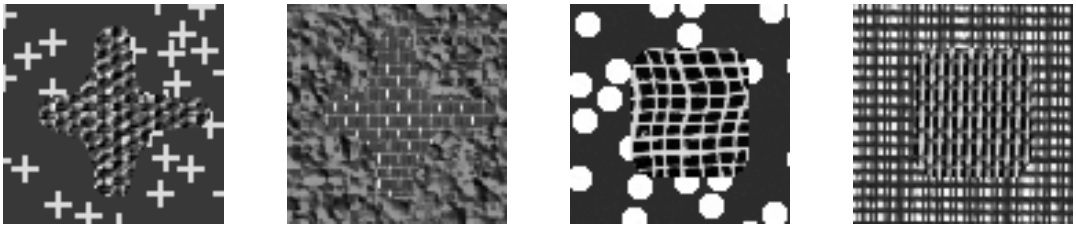
3.5.2.1 Data

We created several texture mosaics of size 80×80 pixels consisting of two regions filled with pairs of the textures used in the experiments above. The mosaics were created by taking 80×80 regions from the original texture images and combining them according to a binary mask. Some examples are shown in Fig. 3.15.

3.5.2.2 Models & Parameters

For the region models p_1, p_2 in equation (3.24) we used the BiFoE texture models learned for the experiments in section 3.4. The mask prior was chosen to be a simple

Masks used to generate texture mosaics



Example mosaics

Figure 3.15: **Test data for the hierarchical, region-based BiFoE** *Top*: Hand-drawn masks (80×80 binary images) used to generate texture mosaics. Each mosaic was generated by extracting a pair of 80×80 patches from two of the original texture images and combining them according to one of the binary masks shown above. *Bottom*: 4 examples of the generated texture mosaics.

4-connected binary MRF (Ising model):

$$p_M(\mathbf{m}) = \exp\left\{ -\beta \sum_{(i,j)\in\mathcal{N}} I[m_i \neq m_j] \right\}, \tag{3.28}$$

where $\mathcal{N}$ is the set of all neighbors in the 4-connected grid and $I[\cdot]$ is the indicator function[5]. In the simulations below we chose $\beta = 4$. We then performed Gibbs sampling for 300 iterations as described in section 3.5.1 to infer the binary mask and the two latent images corresponding to the two textures (the mask was initialized randomly with independent binary noise $p = 0.5$ and the two latent images $\mathbf{x}_1$ and $\mathbf{x}_2$ were initialized with the observed image $\mathbf{y}$).

3.5.2.3 Results

Representative examples of the results for several different texture mosaics are shown in Figure 3.16. The figure shows for each mosaic the test image, the inferred mask, and

[5]The Ising / Potts model is known not to be a good model of region shape (e.g. Morris et al., 1996). We use it here to demonstrate basic principles and will develop richer shape models in chapters 4 and 5.

the two inferred latent images. For the top 5 examples the model is largely successful. It is able to infer the correct state of the mask variables $\mathbf{m}$ for almost all pixels. Furthermore, the model fills in the unseen part of the latent images in a largely plausible manner. Together with the inferred mask, the completed latent image could be used, for instance, to automatically detect and fill-in a hole in a textured region. The bottom three examples show cases where the model has at least partially failed. In some cases it fails to infer the correct segmentation. Such segmentation errors occur primarily where the two textures abut (and where some ambiguity is therefore expected) as well as at the image boundaries where the pixels are less well constrained by the two texture models. In the second to last example the mis-classified pixels could indeed be mistaken to belong to the region filled with texture D4. A second problem that can be observed in some cases is that the model sometimes fails to complete the latent images appropriately, as is the case for the third-last and last row in Fig. 3.16. For the example in the third-last row (textures D77 and D103) this is presumably due to the fact that the visible part of the mesh of texture D103 has a rather extreme conformation and the texture model therefore struggles to complete it in a plausible manner.

3.6 Discussion

We have investigated the FoE's ability to model visual textures. Our results suggest that in its basic form the FoE is a rather limited model of visual structure. Although the filters are learned from data, the zero-centered Student-t potential is too restrictive to model even individual textures. We have further demonstrated that introducing more complex bimodal potentials, and using a better learning strategy, gives rise to a considerably more powerful model. The interactions of multiple bimodal experts can flexibly shape the density allowing the BiFoE to learn good generative, fully parametric models of the textures that we considered.

The results above were obtained with a particular form of a bimodal expert. One interesting question is whether this particular form is crucial. In section 3.4.6 we have discussed the possibility of replacing the bimodal experts with a mixture of two Gaussians with shared $\mathbf{w}_j$ but different biases. Alternatively it might be possible to achieve similar results even with the Student-t potentials when bias terms are included, so that the potentials are no longer necessarily centered at zero (i.e. $\Phi(y;\nu,b) = (1+\frac{1}{2}(y+b)^2)^{-\nu}$; note that the bs are missing in equation 3.10). This formulation could mimick the BiFoE if two experts learned the same filters and ν-

parameters but different bs. In experiments with 1D patterns for which the BiFoE learns a good model without difficulties we found, however, that the FoE with bias terms fails in exactly the same way as the basic FoE (all learned b_js were effectively zero); stable convergence to a bimodal solution (and thus learning a reasonable model) appears extremely difficult to achieve for this model unless the model is effectively initialized with the correct solution.

One interesting question is whether the BiFoE model is not just a more flexible model of specific structure such as textures but also of generic structure in natural images. In preliminary experiments we found that when the model was trained on natural images the filters learned by the BiFoE were qualitatively similar to the ones learned by the basic FoE, and the expert functions were exclusively unimodal and centered at zero. This suggests that, although the BiFoE is a good generative model for specific visual structures, when faced with the task of modeling too heterogeneous a set of patterns (thinking about the structure in a database of natural images as a very large mixture of different textures) it is still not powerful enough and like the basic FoE, accounts only for very generic properties such as smoothness.

Our analysis in 3.4.6 suggests that one of the important differences between the BiFoE and the FoE / GFoE models is the ability of the BiFoE model to represent multi-modal distributions. This raises questions about the shape of the manifolds on which textures live. Is the multi-modal picture suggested by Fig. 3.12 appropriate or are texture manifolds of dramatically different shapes that are still only poorly approximated by our model and would be better represented in other ways?

Interestingly, the importance of modeling the mean has recently been noticed by other authors as well: Ranzato et al. (2010b) evaluate several formulations of MRFs with respect to their suitability as models of generic image structure. Among other models they consider a particular convolutional formulation of the mean-covariance RBM (mcRBM; Ranzato and Hinton, 2010) and a special formulation of the FoE. Collectively the authors refer to these models as "gated MRFs" since they can be thought of as latent variable models in which the covariance of the image pixels is modulated by the state of the latent variables. Consistent with the results discussed in this chapter the authors find that explicitly modeling the mean dramatically improves the generative models learned, and as another similarity to our work they also use SML instead of contrastive divergence which is much used otherwise. Compared to the BiFoE they choose, however, a more flexible formulation in which the mean is modeled by a separate term in the energy, corresponding to a convolutional Gaussian RBM with fixed

variance (cf. section 2.2.4 in chapter 2). This additional flexibility might explain why the considered models lead to noticeably better results on natural images than we have been able to obtain with the BiFoE in our preliminary experiments. The MoG-BiFoE discussed in section 3.4.6 further differs from their models in that the conditional covariance is fixed. As a second improvement they suggest using a "tiled convolutional" formulation, i.e. a model that is not completely stationary but in which parameters are only shared between potentials that are non-overlapping. The resulting, more flexible models in Ranzato et al. (2010b) capture certain aspects of generic image structure and are able to generate large uniform regions with occasional sharp edges. Nevertheless they still fall far short of capturing the diversity in natural images, and, in particular the models seem unable to generate structured, e.g. textured, regions.

In chapter 1 we have argued for a structured approach to image modeling and in section 3.5.2 we have demonstrated how the BiFoE model can be used of a building block of a generative model of textured image regions. With this longer-term goal in mind, there are several directions for future work on the BiFoE model:

- The model is computationally relatively expensive. Efficient auxiliary-variable samplers as used in Welling et al. (2003); Schmidt et al. (2010) are not directly applicable to our model, but alternative formulations of the model might be more amenable to such solutions. Furthermore, a parallel, e.g. GPU-based, implementation would make the model more suitable for large-scale experiments.

- Currently, a separate model has to be trained for each texture. For the model to be a building block of a comprehensive model of general images it would obviously be desirable to have a single model that is able to account for many different textures. Although experiments with one-dimensional patterns suggest that even a single BiFoE model can learn about multiple, different patterns, it is unlikely that a single BiFoE model will be able learn about a very large number of different textures. Also, from a representational perspective it is attractive to have a model that is invariant with respect to different phases or stochastic variations of a *single* texture, but it should also be possible to distinguish between *different* textures. Thus, a hierarchical, latent-variable formulation in which different configurations of the latent variables correspond to different textures appears desirable in this context. In particular, a formulation in which filters are shared between textures and the potential functions are being modulated dependent on the state of latent variables would seem appealing.

- There are many occasions in which texture varies smoothly within a region. One reason why this might happen is, for instance, due to perspective distortions or because the textured surface is not flat. A good generative model of natural image structure should be able to deal with such situations. Furthermore, being able to reason about such texture gradients explicitly could be especially useful since such variations can be a valuable source of information about the physical properties of a scene, such as image depth or the shape of a surface.

In order to employ the texture model in the context of a more comprehensive model of natural images a model of region shape is further required. In the experiments in section 3.5 we have used an Ising model which is known not to be a good model of region shape (e.g. Morris et al., 1996). The question of how to obtain better models of region shape will be the topic of the next two chapters.

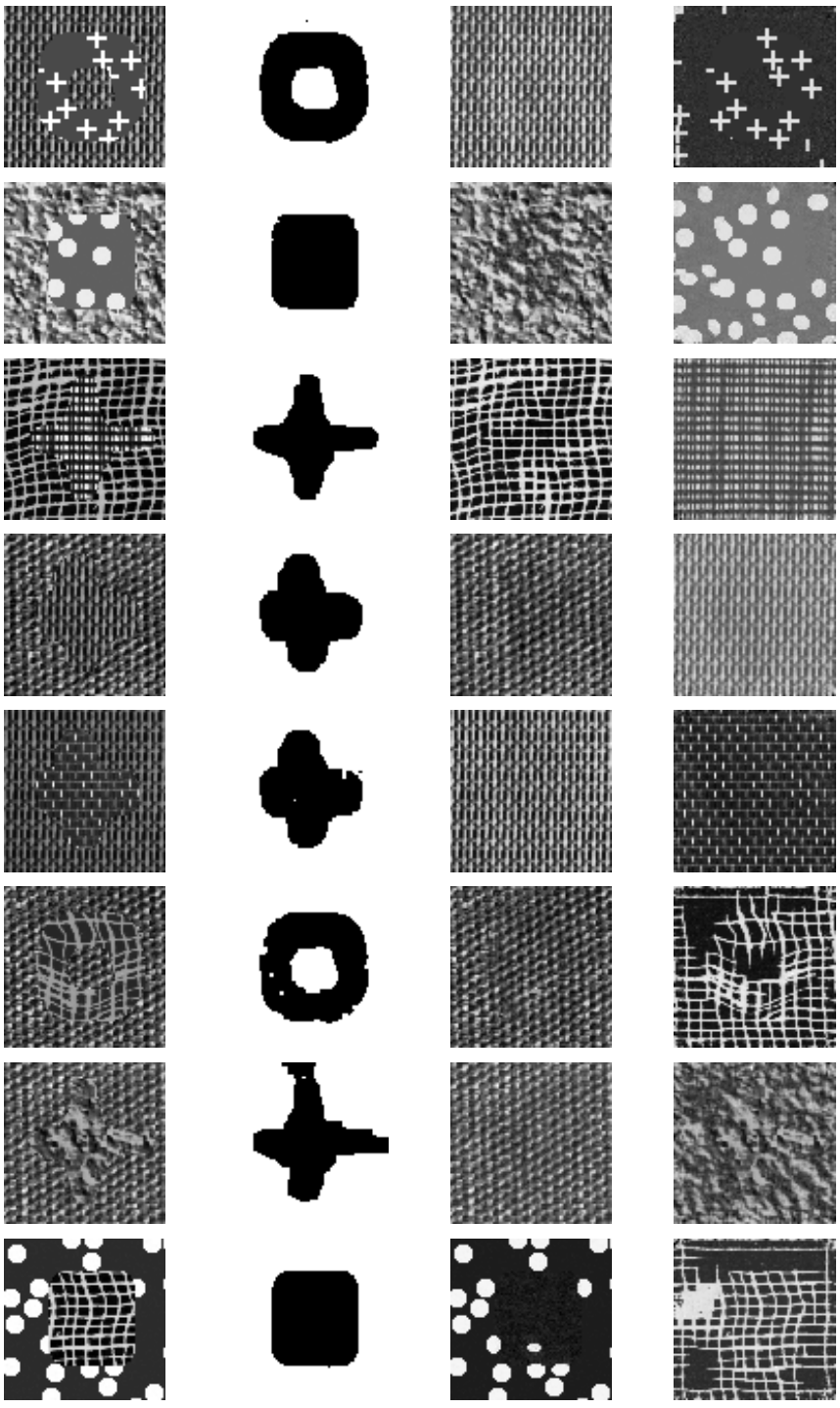


Figure 3.16: **Representative segmentation results from the hierarchical model**. The *first column* shows the 80×80 test images generated as illustrated in Fig. 3.15. The *second column* shows the inferred mask $\mathbf{m}$ after 300 iterations of Gibbs sampling. The *third* and *fourth columns* show the inferred latent images $\mathbf{x}_1$ and $\mathbf{x}_2$. The mask is generally largely correct and the latent images have been completed in a plausible manner. The last three examples illustrate failures of the model.

Chapter 4

Modeling region shape in image patches

4.1 Introduction

In the preceding chapter we have focused on modeling the appearance (texture) of image regions. The work described in this chapter focuses on modeling region *shape*. The shapes of regions in natural images exhibit important regularities (for instance, they are predominantly smooth) and any model of natural images will have to account for these regularities. Knowledge of such regularities is essential not only to generate realistic samples, but also for many image processing tasks such as inpainting, segmentation, or recognition.

The work in this chapter builds on the Masked RBM. The Masked RBM composes images from several regions, and appearances of the regions are modeled as independent draws from a suitable RBM. We endow the Masked RBM model with a model of region shape. The resulting model's generative process can be thought of as composing an image by superimposing several independent objects each of which is represented in terms of its shape and its appearance. The model also incorporates an explicit notion of relative depth and occlusion. Occlusions are an ubiquitous phenomenon in our physical world and it has, for instance, been argued that important aspects of the statistics of natural images can be explained in terms of independent objects occluding each other (Lee et al., 2001). Through the notion of occlusion the model accounts for an important aspect of the image formation process. In particular this allows the model to reason about the true, unoccluded shape of objects, even though most objects are fully visible only very rarely, instead of having to model the vast number of different shapes

that arise from an object being partially occluded.

The idea of modeling the shape of regions in an image by modeling the shapes of individual overlapping objects has previously been pursued in the computer vision literature (see section 4.4). These works, however, have predominantly been limited to sets of relatively homogeneous images (such as images of particular object categories; or the sequences of images that comprise the frames of a movie). In this chapter we will demonstrate how an explicit model of occlusion can be integrated into the deep learning framework. We will show that when using a sufficiently flexible model for object shape this approach can be generalized to model more heterogeneous datasets including natural images.

We will model object shapes using binary RBMs which allow learning of even complicated shape priors. We will show how inference and learning can be implemented efficiently in the model. We will demonstrate that a model that explicitly accounts for occlusions provides a more parsimonious description of the data than alternative models and that, in the context of the masked RBM with a suitable appearance model, it gives rise to powerful model of natural image patches. The work in this chapter will mainly be concerned with relatively small images (i.e. image patches), in chapter 5 we will explain how this model can be extended to images of arbitrary size without making inference prohibitively more expensive.

The remainder of this chapter is structured as follows: Section 4.2 describes the foundation of our work, the Masked RBM developed by our collaborators Nicolas Le Roux and John Winn (Le Roux et al., 2011). Section 4.3 discusses several alternatives to modeling region shape in the context of the masked RBM. We introduce the shape model based on occlusions in section 4.3.2 and discuss inference and learning in section 4.3.3. Some of the most closely related work will be reviewed in section 4.4. Section 4.5.1 describes experiments that demonstrate the general feasibility of learning and inference in the occlusion model and its advantages over alternatives. In section 4.5.2 we show how the occlusion-based shape model gives rise to an interesting model of natural image patches, and we also show that learning about shapes in natural images subsequently allows simple depth inference to be performed in a plausible manner. Section 4.6 summarizes and discusses the work presented in the chapter.

Contribution

The basic formulation of the masked RBM as described in section 4.2 was developed by Nicolas Le Roux and John Winn. My contribution is the occlusion shape model described in section 4.3 as well as the experiments on comparing different shape models and on modeling natural image patches described in section 4.5. The DBN for generating samples from the *appearance* model used to create the samples in section 4.5.2.2 was trained by Nicolas Le Roux. All experiments briefly discussed in section 4.6.2 (Future Work) are fully my own work.

4.2 Masked RBM

In this section we will review the masked RBM as proposed by Nicolas Le Roux and John Winn (cf. Le Roux et al., 2011). The model explicitly accounts for one hallmark of natural images which is the presence of relatively homogeneous regions separated by boundaries. One way to characterize region boundaries is that they represent the breakdown of correlations between neighboring pixels: Pixels that lie on the same side of the boundary are highly correlated whereas pixels that lie on different sides of the boundary are largely independent. When modeling unconstrained natural images region boundaries can appear at arbitrary positions, and there is an extremely large number of alternative appearances for each region. Even when allowing only for the simplest possible case with just a single boundary in any given image (i.e. two regions) and flat colors as "texture" the number of possible combinations rapidly becomes prohibitively large: $(\#\text{colors})^2 \times (\#\text{region boundaries})$. RBMs are capable of modeling high-order correlations between the visible units. The above scenario, however, poses a significant challenge for RBMs conventionally used for image modeling such as the Gauss-Bernoulli RBM described in section 2.2.4 in which the hidden units model only the mean of the visible units: Loosely speaking, such a RBM would need to model explicitly all possible combinations of boundary locations and texture patterns on either side of the boundary. The inefficiency of many simple image models such as the Gauss-Bernoulli RBM when it comes to representing data that is factorial in nature[1] is reflected by the overly smooth samples and reconstructions that are typically generated by such models (see, for instance, the discussion and experiments in Le Roux et al.,

[1] Note that PoE models (cf. section 2.2.3) can represent certain kinds of factorial structure efficiently, a particular challenge in the case at hand is, however, the fact that the correlation structure between image pixels depends on the position of the region boundary.

2011, in particular their Fig. 6).

The masked RBM provides an intuitive way of bypassing this representational inefficiency. An image with K regions is generated by deciding on the appearances of regions independently, and by also generating the shape of the regions independently of their appearances. This idea is illustrated in Fig. 4.1: In the masked RBM an image (patch) with K regions is generated by sampling K images (of the same size as the final image) from a suitable RBM. These image patches determine the appearances of the K regions and are composed according to a "mask" to form the final image. This mask, too, has the same size as the final image and indicates, for each pixel, which of the K regions is visible at that pixel. The pixels of the final image are then set accordingly. In this formulation the RBM that generates region appearances can focus on modeling the consistencies within a region and sharp region boundaries in the final, observed image are generated by switching from one region to the other according to the mask. An additional advantage of this formulation is that it allows reasoning about region shape and appearance explicitly and separately. In the following we will refer to the K patches determining the appearances of the individual regions as "*latent images*" (or patches) and will denote them as $\hat{\mathbf{v}}_k$ ($k \in \{1 \dots K\}$). The final, *observed image* will be denoted by $\mathbf{v}$, and the *mask* by $\mathbf{m}$. If an image is composed from K latent patches we will also say that the model has K layers. If the final, observed image is of dimensionality N and each pixel v_i takes values in $\mathcal{X}$, i.e. $\mathbf{v} \in \mathcal{X}^N$, then the same is true for each latent patch, i.e. $\hat{\mathbf{v}}_k \in \mathcal{X}^N$. The mask is also of dimensionality N, and each element of the mask m_i takes values in $1 \dots K$: $\mathbf{m} \in \{1 \dots K\}^K$ where $m_i = k$ indicates that the k-th region is visible at that pixel, i.e. that $v_i = \hat{\mathbf{v}}_{ki}$.

Below, we will further use the following notation:

- since most of the equations will involve all the layers, we will define a shortcut notation: for any variable t defined for each layer k, the set of variables $\{t_1, \dots, t_K\}$ will be replaced by $t_{1..K}$

- $\mathbf{h}_k^{(a)}$ the hidden state of the k-th layer. The "(a)" superscript stands for "appearance" and distinguishes these hidden units from hidden units for modeling shape that we will introduce later on.

- The masked RBM requires a RBM that models the appearance of individual regions. In principle any RBM can be used for this purpose and its type will primarily depend on the nature of images to be modeled (i.e. on $\mathcal{X}$). In the

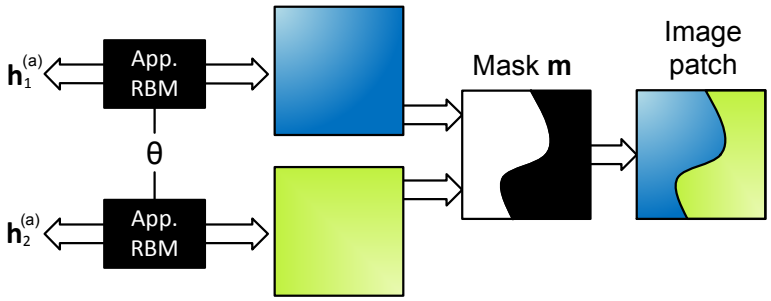


Figure 4.1: **The Masked RBM**. A masked RBM models an image patch as the composition of two or more latent patches, each generated from a separate appearance RBM with shared parameters θ. The composition is controlled by a mask $\mathbf{m}$, indicating which of the latent image patches is to be used to model each visible image pixel. Figure courtesy of Nicolas Le Roux, John Winn & Jamie Shotton.

experiments on modeling natural image patches below we will use a particular continuous valued RBM as described in section 4.5.2.1. For now we will just generically denote the joint distribution over visible and hidden variables $(\widehat{\mathbf{v}}_k, \mathbf{h}_k^{(a)})$ defined by the chosen RBM as $\mathrm{APP}(\widehat{\mathbf{v}}_k, \mathbf{h}_k^{(a)})$ and the conditional distributions as $\mathrm{APP}(\widehat{\mathbf{v}}_k|\mathbf{h}_k^{(a)})$ and $\mathrm{APP}(\mathbf{h}_k^{(a)}|\widehat{\mathbf{v}}_k)$ respectively. Furthermore, since the conditional distributions factorize we will write for instance $\mathrm{APP}(\hat{v}_{k,i}|\mathbf{h}_k^{(a)})$ to denote the conditional distribution over pixel i given the state of the hidden units (and accordingly for the conditional distribution over individual hidden units).

Using these notations, and given a mask $\mathbf{m}$, the probability of a joint state $S = \{\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}_{1..K}^{(a)}\}$ is equal to

$$P(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}_{1..K}^{(a)}|\mathbf{m}) = \left(\prod_i \delta\left[\widehat{v}_{m_i,i} = v_i\right]\right)\left(\prod_k \mathrm{APP}(\widehat{\mathbf{v}}_k, \mathbf{h}_k^{(a)})\right). \tag{4.1}$$

The first term assigns zero probability to configurations violating the constraint that, if layer k is selected to explain pixel i (i.e. $m_i = k$), then we must have $\widehat{v}_{k,i} = v_i$. Fig. 4.2 shows the factor graph associated with this model. This model is a chain graph: Although the latent images $\mathbf{v}_k$ individually are generated from an undirected graphical model (RBM) they are marginally independent and given the mask $\mathbf{m}$ the image is composed in a deterministic manner corresponding to a directed graphical model as depicted schematically in the top half of Fig. 4.3.

As described in Le Roux et al. (2011), assuming a prior over the mask that factorizes in a suitable manner with respect to the image pixels inference can be performed

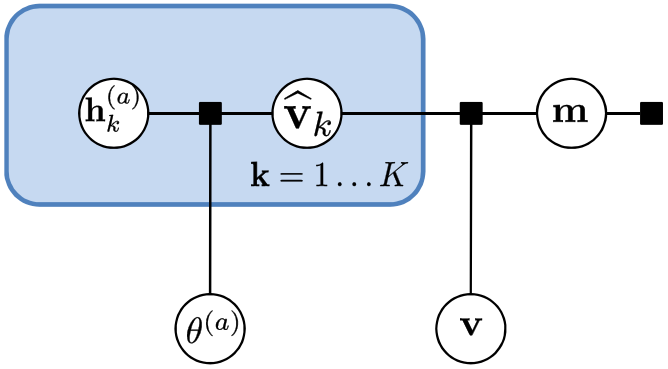


Figure 4.2: **Factor graph of the masked RBM without mask prior.** The joint distribution between the latent images $\widehat{\mathbf{v}}_k$ and corresponding hidden units $\mathbf{h}_k^{(a)}$ is modeled by an RBM with parameters $\theta^{(a)}$. $\theta^{(a)}$ is outside the plate and thus the same for all RBMs. The latent images are composed with a mask $\mathbf{m}$ to form the image patch $\mathbf{v}$. The graphical model includes a factor to generically indicate a prior over the mask. In the simplest case this can just be a uniform distribution, but more interesting priors will be discussed in section 4.3.

efficiently through a Gibbs sampling scheme which is given in full in the appendix (section B.1). Many variables can be updated independently because, given the latent variables $\mathbf{h}_{1..K}^{(a)}$, the distribution over the mask variables and the visible units factorize with respect to the pixels:

$$P(\widehat{\mathbf{v}}_{1..K}, \mathbf{m} | \mathbf{v}, \mathbf{h}_{1..K}^{(a)}) = \prod_i P(\hat{v}_{1,i} \ldots \hat{v}_{K,i}, m_i | v_i, \mathbf{h}_{1..K}^{(a)}), \tag{4.2}$$

where, for simplicity, we have assumed a uniform prior over the mask. This factorization is a result of the fact that the joint probability of a RBM can be written to factorize with respect to the visible units (cf. equation (2.28) in section 2.2.4), and the same is true for the product of delta functions in equation (4.1).

Furthermore, this distribution can be decomposed as follows:

$$P(\widehat{\mathbf{v}}_{1..K}, \mathbf{m} | \mathbf{v}, \mathbf{h}_{1..K}^{(a)}) = \prod_i P(m_i | v_i, \mathbf{h}_{1..K}^{(a)}) \left(\prod_k P(\hat{v}_{k,i} | v_i, m_i, \mathbf{h}_{1..K}^{(a)}) \right) \tag{4.3}$$

$$P(m_i = k | v_i, \mathbf{h}_{1..K}^{(a)}) \propto \mathrm{APP}(v_i | \mathbf{h}_k^{(a)}). \tag{4.4}$$

Given the complete latent images $\widehat{\mathbf{v}}_{1..K}$ the latent variables $\mathbf{h}_{1..K}^{(a)}$ can also be sampled efficiently, sampling from $P(\mathbf{h}_k^{(a)} | \widehat{\mathbf{v}}_{1..K}, \mathbf{m}) = \mathrm{APP}(\mathbf{h}_k^{(a)} | \widehat{\mathbf{v}}_k) = \prod_j \mathrm{APP}(h_{k,j}^{(a)} | \widehat{\mathbf{v}}_k)$. Note, however, that given a mask some pixels in the latent images $\widehat{\mathbf{v}}_{1..K}$ are unobserved as shown in Fig. 4.3. It would be desirable to integrate out these unobserved latent pixels

but this is not possible.[2] Instead, these are "filled in" by performing blocked Gibbs sampling between the unobserved latent pixels $\hat{\mathbf{v}}_{k,i:m_i \neq k}$ and the hiddens $\mathbf{h}_k$ conditioned on the observed latent pixels $\hat{\mathbf{v}}_{k,i:m_i=k}$ as explained in section B.1 in the appendix.

Le Roux et al. (2011) investigate the masked RBM as described above and find that it is indeed considerably better at representing sharp region boundaries and, more generally, high-frequency structure than conventional RBMs. They further find that when trading off the number of layers K against the capacity of the appearance RBM it is advantageous to choose a larger K and a smaller number of hidden variables (see their Figure 6).

[2] For certain choices of APP it would be possible to integrate out the unobserved latent pixels, but this would introduce high-order dependencies between the hidden units so that they would no longer be conditionally independent (and block Gibbs sampling thus impossible).

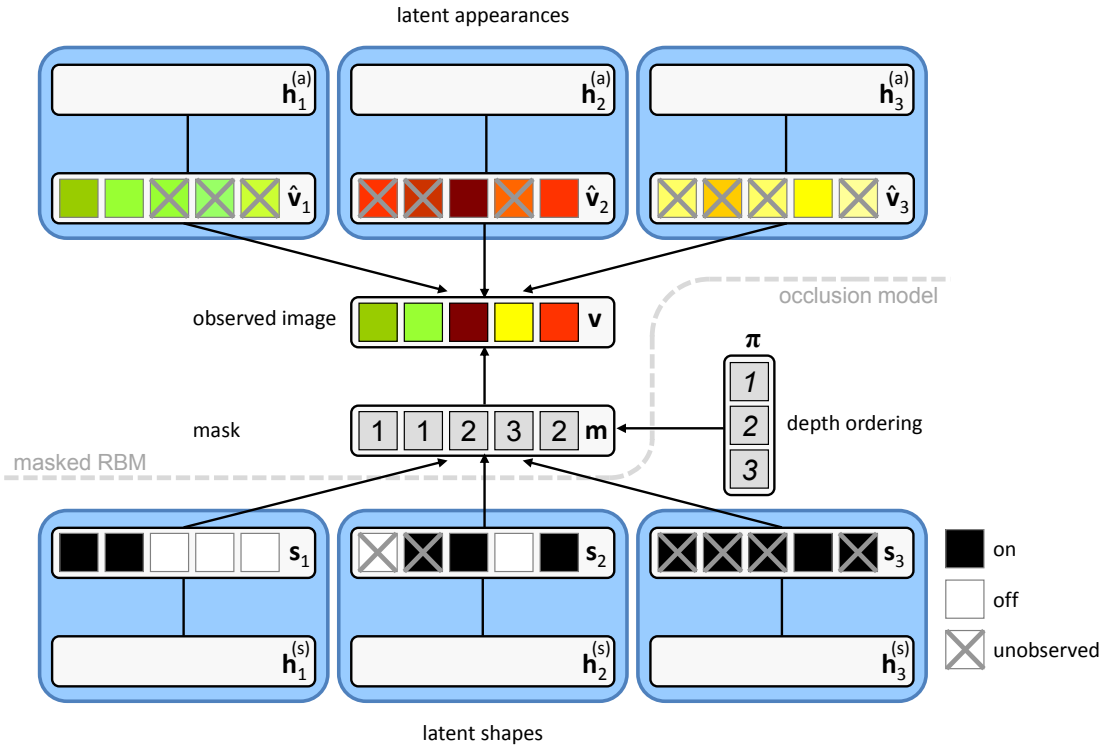

latent appearances
$\mathbf{h}_1^{(a)}$
$\mathbf{h}_2^{(a)}$
$\mathbf{h}_3^{(a)}$
$\hat{\mathbf{v}}_1$
$\hat{\mathbf{v}}_2$
$\hat{\mathbf{v}}_3$
observed image
$\mathbf{v}$
occlusion model
$\boldsymbol{\pi}$
1
2
3
depth ordering
mask
1 1 2 3 2 $\mathbf{m}$
masked RBM
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_3$
$\mathbf{h}_1^{(s)}$
$\mathbf{h}_2^{(s)}$
$\mathbf{h}_3^{(s)}$
on
off
unobserved
latent shapes

Figure 4.3: **Schematic of the masked RBM with occlusion mask model.** Specialization of the general factor graphs from Figures 4.2 and 4.5 as a chain graph-like schematic. The schematic is shown for the full masked RBM with occlusion mask prior and $K = 3$ as described in sections 4.2 and 4.3.2. The upper part of the figure (above the dashed line, labeled as "*masked RBM*") corresponds to the model described in section 4.2, i.e. the basic formulation of the masked RBM without explicit model for the mask, the factor graph of which is shown in Fig. 4.2. The lower part of the figure (below the dashed line, labeled as "*occlusion model*") is a schematic of the occlusion mask model described in section 4.3.2. The full figure corresponds to the joint model (shape and appearance) whose factor graph is shown in Fig. 4.5. Unlike the factor graphs this figure distinguishes between undirected and effectively directed interactions between variables. In addition to the model structure the figure also shows a particular instantiation of all variables involved. In the *masked RBM* (upper part of the figure) three latent appearances $\hat{\mathbf{v}}_{1...3}$ are drawn from a suitable RBM (with visible units $\hat{\mathbf{v}}$ and hidden units $\mathbf{h}^{(a)}$). Given a state of the mask $\mathbf{m}$ these are combined to form the observed image $\mathbf{v}$: for instance, $m_1 = 1$ and v_1 is therefore set to the value of $\hat{v}_{11}$, which is green. For each latent appearance only some pixels are visible in the observed image, others are *unobserved* as indicated by the crosses. As described in section 4.3.2, in the occlusion shape model (lower part of the figure) the mask $\mathbf{m}$ is composed from K binary shapes $\mathbf{s}_{1...3}$ drawn from the shape RBM (with visible units $\mathbf{s}$ and hidden units $\mathbf{h}^{(s)}$. These binary shapes are combined according to the depth order π in an occluding manner to form the mask: For instance, $k = 1$ is the front most layer ($k = 2$ is in the middle and $k = 3$ is rear-most), and since $s_{11} = 1$ the mask pixel m_1 is set to $m_1 = 1$. As for the appearances not all shape pixels are observed, e.g. s_{21} and s_{31} are both unobserved since layer 2 and 3 are behind layer 1 and $s_{11} = 1$. $s_{23} = 1$, however, is visible since $s_{13} = 0$, and therefore $m_3 = 2$. Note that the K latent appearances and shapes are collapsed into a single plate respectively in the factor graphs in Figures 4.2 and 4.5. Note further that that hidden units of the shape and appearance RBMs are not shown individually.

4.3 Modeling shape and occlusion

One important question that is left open by the formulation of the masked RBM presented in the previous chapter, is how to model the shape of the image regions, i.e. the question of modeling the mask. Eq. (4.1) defines a conditional distribution of the image, the latent patches and the hidden states *given* the mask. To get a full probability distribution over the joint variables, we must also define a distribution over the mask. The mask is effectively an image, where each pixel can take on one out of K values. In this section we will discuss various possibilities to model such K-valued images. In particular, in section 4.3.2 we will propose a model in which the partition of the image into regions is obtained by generating several independent shapes that are then arranged in an occluding manner so that the image regions arise as visible parts of the occluding shapes. Before discussing the occlusion model, however, we will first discuss two alternative models that are considerably simpler and help to motivate the occlusion-based model. These latter two models are the uniform model, which assumes that all states of the mask are equally likely, and a simple multinomial RBM, which we will refer to as the "softmax" model. One important consideration to keep in mind for the remainder of this section is that we are aiming to develop models for images with K regions where all K regions are equivalent. This is in contrast to other models such as the ones that will be discussed in section 4.4.2 which assume that an image is composed from regions that differ in their characteristics (e.g. composing an image from several foreground objects with different characteristics and a background) and thus should be governed by different models.

4.3.1 Simple shape models: Uniform and Softmax

4.3.1.1 Uniform mask model

The simplest mask model is the uniform distribution over $\mathbf{m}$. In this model, no particular state of the mask is preferred a priori, i.e. $p(\mathbf{m}) = \frac{1}{K^N}$, which is what we have assumed in equations (4.2-4.4). While, from a generative perspective this model makes little sense, it can still be used during inference, i.e. to segment an image using the masked RBM. The segmentation of the image into different "regions" is, however, purely driven by the image, and segmentations in which pixels of different regions are highly interleaved are a priori equally likely to strongly coherent segmentations. In many cases this leads to very noisy segmentations as is demonstrated e.g. in Le Roux

et al. (2011). We use this model as a baseline.

One simple prior to encourage mask-regions to be coherent would be the Ising model (for binary masks) or its generalization the Potts model (for K-valued masks), a simple pairwise MRF which we have discussed in section 2.2.1.1 and employed in the context of the region-based BiFoE model in section 3.5. This model is, however, a rather limited model of region shape (Tjelmeland and Besag, 1998; Morris et al., 1996) and inference in this model would be computationally much less efficient since the mask pixels would not be conditionally independent. A more promising choice is the multinomial RBM which will be discussed next.

4.3.1.2 Softmax model

The softmax model for K-valued N-dimensional images is a multinomial RBM (see section 2.2.4) with N visible units. Each of these units takes on values $1 \dots K$. Alternatively it can be thought of as having K sets of N visible binary units $\mathbf{s}_{1..K}$, i.e. K binary units per pixel. For each pixel, only one of the K units can be turned on. The k-th unit being on for pixel i (i.e. $s_{k,i} = 1$, and thus also $s_{k,i} = 0 \quad \forall k' \neq k$) corresponds to pixel i having value k: $s_i = k$.

As explained above we consider all layers (or regions) in an image to have equal status (i.e. we do not, for instance, distinguish between foreground and background). We therefore consider a special form of the softmax model which consists of K binary RBMs with shared parameters competing to explain each mask pixel. One possible interpretation of this model is that each RBM defines a joint distribution over its visible $\mathbf{s}_k$, which specify a binary shape, and its binary hidden units $\mathbf{h}_k^{(s)}$ (the "(s)" superscript stands for "shape"). The K binary shapes $\mathbf{s}_k$ are then combined to form the mask $\mathbf{m}$, which is a K-valued vector of the same size as the $\mathbf{s}_k$'s. To determine the value of m_i given the K sets of hidden states $\mathbf{h}_k^{(s)}$, one needs to compute a softmax over the K different inputs. The joint probability distribution of this model is:

$$P(\mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}_{1..K}^{(s)}) \propto \left(\prod_i \delta(s_{m_i,i} = 1) \prod_{k \neq m_i} \delta(s_{k,i} = 0) \right) \left(\prod_k \text{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)}) \right) \quad (4.5)$$

where $\text{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)})$ is the joint probability of $(\mathbf{s}_k, \mathbf{h}_k^{(s)})$ under the chosen shape RBM (a binary RBM in our case). The right-hand side of the equation is unnormalized since not all configurations of the visibles $\mathbf{s}_{1...K}$ give rise to a valid mask ($s_{k,i} = 0$ for all k, for instance).

The first and second terms state that exactly one shape has to be "on" at any given pixel, and that the index of the selected shape is the value of the mask at that pixel. This constraint introduces a coupling between the K shape models. Unlike the appearances in eq. (4.1) the shapes are not independent. An alternative way to think about the above model is in terms of a single multinomial RBM with certain constraints on the connectivity and weight sharing.

Inference: The advantage of the model being undirected in nature and of the shapes being fully observed is that inference can be implemented efficiently as it allows for straightforward blocked Gibbs sampling:

$$
\begin{aligned}
P(\mathbf{s}_{1..K}|\mathbf{h}^{(s)}_{1..K}) &= \prod_i P(s_{1,i},\dots s_{K,i}|\mathbf{h}^{(s)}_{1..K}), && (4.6)\\
P(m_i = k|\mathbf{h}^{(s)}_{1..K}) &= P(s_i = k|\mathbf{h}^{(s)}_{1..K}) \\
&\propto \text{SHAPE}(s_{k,i} = 1|\mathbf{h}^{(s)}_k), && (4.7)\\
P(\mathbf{h}^{(s)}_{1..K}|\mathbf{s}_{1..K}) &= \prod_k \text{SHAPE}(\mathbf{h}^{(s)}_k|\mathbf{s}_k) && (4.8)\\
&= \prod_{k,j} \text{SHAPE}(h^{(s)}_{k,j}|\mathbf{s}_k). && (4.9)
\end{aligned}
$$

Note that equation (4.7) gives rise to the softmax activation function:

$$
P(s_i = k|\mathbf{h}^{(s)}_{1..K}) = \frac{\exp(W^{(s)}_{i\cdot}\mathbf{h}^{(s)}_k)}{\sum_{k'}\exp(W^{(s)}_{i\cdot}\mathbf{h}^{(s)}_{k'})} \tag{4.10}
$$

where $W^{(s)}$ are the weights of the binary shape RBM (see also section 2.2.4 in chapter 2; note that since the parameters are shared across layers the visible biases cancel out)

Learning: Learning can be implemented efficiently in this model using any of the learning criteria commonly used for RBMs (cf. chapter 2). Given a set of training mask images $\mathbf{m}^{(1)}\dots\mathbf{m}^{(T)}$ the contrastive divergence update is, for instance, obtained as follows: For each mask image $\mathbf{m}^{(t)}$ we sample $\mathbf{h}^{(t)+}_{1\dots K}$ according to $\text{SHAPE}(\mathbf{h}^{(t)+}_k|\mathbf{s}^{(t)}_k)$ (using eq. 4.9) for the positive part of the gradient. We then obtain samples $\mathbf{s}^{(t)-}_{1\dots K}$, $\mathbf{h}^{(t)-}_{1\dots K}$ for the negative part of the gradient by performing Gibbs sampling according to equations (4.6) – (4.10) starting either from the data (for contrastive divergence; CD) or from the current particles in the persistent chains (for stochastic maximum likelihood; SML) (note that this effectively involves sampling full mask images $\mathbf{m}$ when sampling

the binary shapes given the states of the hidden units)[3]. We then compute

$$\Delta\theta \propto -\frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\left(\frac{\partial}{\partial\theta}E_s\left(\mathbf{s}_k^{(t)},\mathbf{h}_k^{(t)+}\right)-\frac{\partial}{\partial\theta}E_s\left(\mathbf{s}_k^{(t)-},\mathbf{h}_k^{(t)-}\right)\right), \tag{4.11}$$

where E_s is the energy of the binary RBM SHAPE as described in chapter 2, parameterized by the weights $W^{(s)}$, and hidden biases $\mathbf{c}^{(s)}$. $\mathbf{s}_{1\ldots K}^{(t)}$ are the binary vectors corresponding to the state of the t-th mask image $\mathbf{m}^{(t)}$.

Its simplicity makes this model appealing. The downside, however, is that the model poorly reflects the process of natural image formation. In particular, this model is not able to handle occlusions properly: due to the constraints in equation (4.5) if one object is present at pixel i, then none of the other objects can be. Thus, when object A is occluding object B, the shape of object B is considered as absent in the occluded region rather than unobserved. One way to think about this is that the model makes the implicit assumption that all the objects are at the same depth. As a consequence, the model is forced to learn the shape of the visible regions of occluded objects instead of their true (unoccluded) shape. This idea is illustrated in Fig. 4.4 for two mask images that show a circle partially occluding a square and *vice versa*. In order to model these mask images the softmax model needs to be able to generate a square and a circle, but also a square and a circle with the occluded parts missing. The reason for this is the nature of the softmax activation function given in equation (4.10): In order to reliably turn on unit $s_{i,k}$ its input $W_{i\cdot}^{(s)}\mathbf{h}_k^{(s)}$ needs to be larger than the input to all other units $s_{i,k'}$ $(k' \neq k)$. One consequence of this is that there will be no direct correspondence between the hidden states of any single layer and the corresponding object shape, since the observed shape will jointly depend on the K inputs. In an object recognition system, this is likely to reduce the ability to recognize partially occluded objects based on their shape.

4.3.2 The occlusion model

An occlusion occurs when an object is partially hidden by some other object. In this case the visible shape of the object does not correspond to the true shape of the object. As explained, the softmax model cannot represent this situation since the shape of the mask regions and the shape of the underlying objects are tied. In the occlusion model we take into account the fact that objects can occlude each other by introducing an

[3]For instance, for CD-1 we first sample $\mathbf{s}_{1\ldots K}^{(t)-} \sim P(\cdot|\mathbf{h}_{1\ldots K}^{(t)+})$ in eq. (4.10) (this step corresponds to sampling a mask image $\mathbf{m}$), and then $\mathbf{h}_k^{(t)-} \sim \text{SHAPE}(\cdot|\mathbf{s}_k^{(t)-})$.

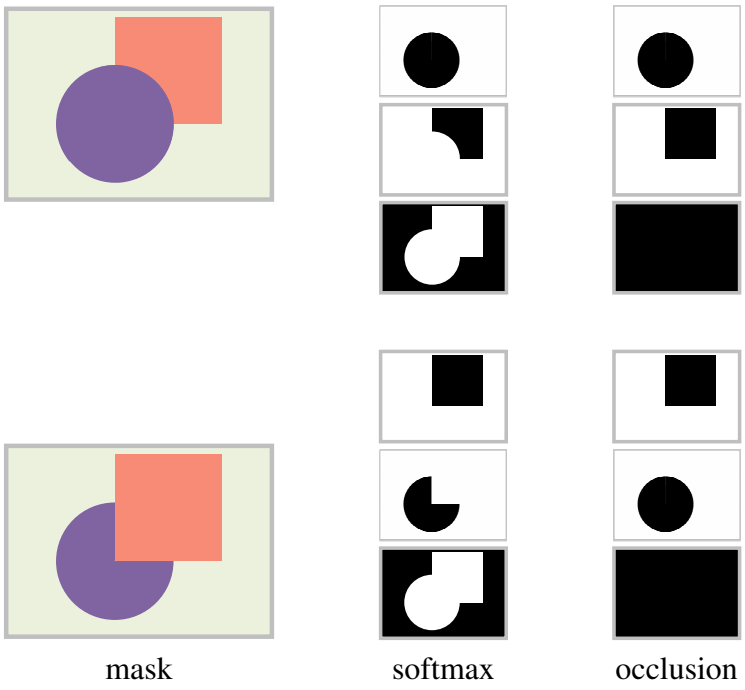


Figure 4.4: **Modeling region shape: Softmax vs. Occlusion model.** Illustration of the difference between the shape representation in the softmax and the occlusion model for two mask images. Both mask images contain a circle and a square in front of a homogeneous background. The softmax model implicitly assumes that all shapes are at the same depth, i.e. for one object to be present at one particular image location all other objects have to be absent. Thus, whereas 5 different shapes are required to represent the two mask images in the softmax model (circle, square, circle with cut-out, square with cut-out, background with cut-out), both mask images can be generated from only three different shapes in the occlusion model (circle, square, background).

explicit depth ordering. This allows modeling the true (unoccluded) shape of objects but to also generate mask images in which objects are only partially visible.

We assume that the K layers containing the shapes of the objects that will form the mask image are arranged according to a depth ordering π. $\pi(k)$ is the position in the relative depth ordering of layer k, i.e. $\pi(k) = 1$ indicates that k is the front-most layer and $\pi(k) = K$ indicates that k is the rear-most layer[4]. For an object (shape) to

[4]Below we will also use $\pi^{-1}(\cdot)$ which, for a given depth value, returns the index of the layer at that depth, so that, for instance, $\pi^{-1}(1)$ returns the front most layer, and $\pi^{-1}(K)$ returns the index of the rear-most layer.

be visible, there must not be any other shape at the same location in the layers above. This idea is illustrated in Fig. 4.4 (right column). Unlike in the softmax model the two mask images containing a circle in front of a square and the square in front of a circle can be represented with only three different shapes (the square, the circle, and the background).

The joint probability distribution for this model can be written as follows:

$$P(\mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi) \propto \quad P(\pi) \quad \left(\prod_i \delta(s_{m_i,i} = 1) \prod_{k:\pi(k)<\pi(m_i)} \delta(s_{k,i} = 0) \right) \times \quad \left(\prod_k \text{SHAPE}(\mathbf{s}_k, \mathbf{h}^{(s)}_k) \right) \tag{4.12}$$

The joint distribution is defined over four sets of variables: the mask $\mathbf{m}$ (reflecting the visible part of each shape), the latent shapes $\mathbf{s}_{1\ldots K}$ (reflecting the true, unoccluded shapes), the corresponding hidden units $\mathbf{h}^{(s)}_{1\ldots K}$, and the depth ordering π. The distribution consists of three components:

1. A prior over the possible depth orderings $P(\pi)$. As explained in section 4.2 we assume that all layers are a priori equal so there is no reason to prefer one ordering over another. We therefore chose $P(\pi)$ to be uniform in the rest of the thesis.

2. A binary RBM, $\text{SHAPE}(\mathbf{s}_k, \mathbf{h}^{(s)}_k)$, for each layer $k = 1 \ldots K$. These RBMs model the true (unoccluded) shapes of the K objects that comprise an image. Again, since we assume that all regions are a priori equal we use the same shape model for all K regions (note that each RBM has its own set of latent variables, but the parameters are shared).

3. A term (product of delta functions) that encodes the occlusion constraint.

The model looks structurally similar to the softmax model described in the previous section but it has rather different properties. It is the term that encodes the occlusion constraint that is responsible for the crucial difference to the softmax model. It decouples the object shape from the shape of the corresponding image region: In equation (4.12) if $m_i = k$, then we must have $\mathbf{s}_{k,i} = 1$ (i.e. the visible shape needs to be on, as in the softmax model), but we only require that $\mathbf{s}_{k',i} = 0$ for the layers k' in front of the layer k (i.e. only the shapes in the layers in front of the visible layer have to be off, rather than for all the layers as is the case for the softmax model). $\mathbf{s}_{k'',i}$ for k'' behind

layer k are unobserved (occluded). This idea is illustrated in Fig. 4.6 and has two important consequences: Firstly, the shape model is now free to focus on modeling the true shapes instead of the shape of the visible regions which is likely to allow for a more efficient representation. Secondly, this should admit for a more direct correspondence between the hidden states and the shapes of the objects present in an image.

The general factor graph corresponding to the masked RBM with non-uniform mask prior is shown in Fig. 4.5. Figure 4.3 specializes this general factor graph for the masked RBM with occlusion mask model and shows a schematic of the full model as a chain graph. There is a notable symmetry between the shape and appearance part of the model. In particular, when generating a new image both shape and appearance will first be drawn independently for each layer, and will then be composed to form a mask and the full image.

There is one subtlety to the above formulation of the occlusion model, which is related to the proportionality sign in equation (4.12): The right hand side of this equation is unnormalized due to configurations of the visibles violating the constraints. This occurs when all shapes are off at a particular pixel, i.e. $s_{k,i} = 0$ for all k. When generating a mask from the occlusion mask model this could be dealt with by rejecting such invalid shape tuples. This corresponds to a re-normalization of eq. (4.12) and means that the shapes are not truly marginally independent. In practice we therefore take a different approach: when generating from the occlusion model we do not draw the shape for the rear-most layer from the shape RBM but rather assume that this layer's shape is always on everywhere where it is visible, i.e. for all pixels that are not covered by any of the preceding shapes (shapes in layers $k : \pi(k) < K$). This can be thought of as drawing the rear-most shape from a special shape model that puts all probability mass at the fully filled shape and the generative model remains thus well defined. In this view, eq. 4.12 does not include the term $\mathrm{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)})$ for $k = \pi^{-1}(K)$ (i.e. for the rear-most shape). This formulation is still a well-defined model, the joint distribution is normalized and the shapes are marginally independent, giving rise to the directed edges in Fig. 4.3.

The full generative process can thus be summarized as follows:

1. Sample a depth ordering π
2. For each layer $k = 1 \ldots K$ generate an appearance $\hat{\mathbf{v}}_k$ by drawing K independent samples from the appearance RBM.
3. For each layer $k = 1 \ldots K$ generate a shape $\mathbf{s}_{1 \ldots K}$ by drawing independent samples

from the shape RBM.

4. Generate the mask $\mathbf{m}$ using the sampled shapes $\mathbf{s}_{1\ldots K}$ and the depth ordering π.

5. Compose the image $\mathbf{v}$ using the mask $\mathbf{m}$ and the appearances $\hat{\mathbf{v}}_{1\ldots K}$.

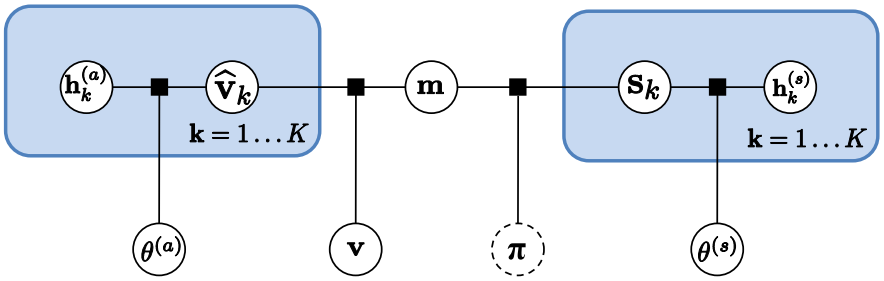


Figure 4.5: **Factor graph of the masked RBM with a non uniform mask prior.** The joint distribution between the shapes $\mathbf{s}_k$ and the hidden shape states $\mathbf{h}_k^{(s)}$ are modeled by an RBM with parameters $\theta^{(s)}$. $\theta^{(s)}$ is outside the plate and thus the same for all RBMs. The ordering π is only used in the occlusion model.

4.3.3 Inference & learning in the occlusion model

4.3.3.1 Inference

Perhaps surprisingly, the occlusion-based shape model admits efficient inference based on blocked Gibbs sampling. For this, the following properties of the distribution defined in (4.12) are important:

1. Given the latent shapes $\mathbf{s}_{1\ldots K}$ the distribution over the shape hidden states $\mathbf{h}_{1\ldots K}^{(s)}$ is factorial (this is simply due to the fact that each shape is modeled by an RBM):

$$P(\mathbf{h}_k^{(s)}|\mathbf{s}_k) = \prod_j \text{SHAPE}(h_{k,j}|\mathbf{s}_k) \tag{4.13}$$

2. Given the mask $\mathbf{m}$, the shape hidden states $\mathbf{h}_k^{(s)}$ and the ordering π the distribution over the latent shapes $\mathbf{s}_{1\ldots K}$ is factorial:

$$P(\mathbf{s}_k|\mathbf{m},\mathbf{h}_k^{(s)},\pi) = \prod_i P(\mathbf{s}_{k,i}|\mathbf{m},\mathbf{h}_k^{(s)},\pi) \tag{4.14}$$

$$P(\mathbf{s}_{k,i}|\mathbf{m},\mathbf{h}_k^{(s)},\pi) = \begin{cases} \delta(s_{k,i}=1) & \text{if } m_i = k \\ \delta(s_{k,i}=0) & \text{if } \pi(k) < \pi(m_i) \\ \text{SHAPE}(\mathbf{s}_{k,i}|\mathbf{h}_k^{(s)}) & \text{otherwise} \end{cases} \tag{4.15}$$

This follows directly from equation. (4.12) and the fact that SHAPE is a RBM and the conditional distribution over the visible units is therefore factorial.

3. As shown in Appendix B.2 the distribution over mask pixels is factorial given the shape hidden states $\mathbf{h}_k^{(s)}$ and the depth ordering π:

$$P(m_i = k|\mathbf{h}_{1..K}^{(s)}, \pi) \propto \text{SHAPE}(s_{k,i} = 1|\mathbf{h}_k^{(s)}) \times \prod_{k':\pi(k')<\pi(k)} \text{SHAPE}(s_{k',i} = 0|\mathbf{h}_k^{\prime(s)}). \quad (4.16)$$

Given a depth ordering, these three properties suggest a Gibbs sampling scheme in which we sample the shape hidden units $\mathbf{h}_{1...K}^{(s)}$ given the latent shapes $\mathbf{s}_{1...K}$, the mask $\mathbf{m}$ given the state of the shape hidden units, and the latent shapes given the mask and the shape hidden units. Importantly, in the full model, i.e. when the occlusion shape model is used as a prior over the mask in the masked RBM we can simply combine the signal given by (4.16) with the signal from the appearance model given by (4.3) when re-sampling the mask. This is described in more detail in section 4.3.4.

All the above steps were conditioned on a depth ordering π. In order to infer the depth variable π given a mask $\mathbf{m}$, we consider each possible ordering of the K layers explicitly. The mask $\mathbf{m}$ together with a particular occlusion order π defines which shape pixel $s_{k,i}$ are observed and which are unobserved. This is illustrated in Fig. 4.6. The likelihood of a particular ordering π is then simply given as the likelihood of all the partially observed shapes $\mathbf{s}_k$ under the shape model:

$$P(\pi|\mathbf{m}) \propto \prod_{k=1}^{K} \sum_{\{s_{k,i}:i\in U_{\pi,k}(\mathbf{m})\}} \sum_{\mathbf{h}_k^{(s)}} \text{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)}), \quad (4.17)$$

where the first sum is over the unobserved shape pixels: $U_{\pi,k}(\mathbf{m})$ is the set of all unobserved pixels for shape k given the mask $\mathbf{m}$ and the ordering π. The set of unobserved pixels $U_{\pi,k}(\mathbf{m})$ will vary between different orderings π and this is what drives the depth inference.

In practice the sum over unobserved pixels *and* over the latent variables $\mathbf{h}_k^{(s)}$ cannot be computed exactly. We therefore replace the first sum by sampling the unobserved pixels $\{s_{k,i} : i \in U_{\pi,k}(\mathbf{m})\}$ conditioned on the observed shape pixels for each k and π. Sampling can be done efficiently using several iterations of block Gibbs sampling using equations (4.13) and (4.15). This results in "completed" shape images $\hat{s}_k^{\pi}$ for which the unnormalized probability under the shape model can be computed efficiently

as described in section 2.2.4 (equation (2.27))

$$p(\hat{\mathbf{s}}_k^\pi) = \sum_{\mathbf{h}} \text{SHAPE}(\hat{\mathbf{s}}_k^\pi, \mathbf{h}) \tag{4.18}$$

$$\propto \exp\left(\left(\mathbf{b}^{(s)}\right)^T \hat{\mathbf{s}}_k^\pi\right) \prod_j \left[1 + \exp\left((\hat{\mathbf{s}}_k^\pi)^T W_{\cdot j}^{(s)} + c_j^{(s)}\right)\right], \tag{4.19}$$

so that we obtain

$$P(\pi|m, \hat{\mathbf{s}}_{1..K}^\pi) \propto \prod_{k=1}^{K} \text{SHAPE}(\hat{\mathbf{s}}_k^\pi). \tag{4.20}$$

It is important to realize that the completed shape images will be different for different π (therefore the superscript in $\hat{\mathbf{s}}_k^\pi$): For plausible orderings, the shape model will be able to "fill in" the unobserved pixels to give rise to a shape with a high likelihood, which in turn leads to a high probability of the respective ordering (cf. Fig. 4.6). Considering each possible ordering π explicitly might seem expensive (the number of possible orderings is factorial in K), but it remains feasible in practice for $K \leqslant 4$.

The shape in the rear-most layer is largely determined by the preceding layers. For this reason, and as explained in section 4.3.2, we treat the rear-most shape in a special manner. During depth inference this means that we ignore the likelihood of the rear-most shape when computing the probability of a particular depth ordering π using eq. (4.20), i.e. $P(\pi|m, \hat{\mathbf{s}}_{1..K}^\pi) \propto \prod_{k:\pi(k)\neq K} \text{SHAPE}(\hat{\mathbf{s}}_k^\pi)$. Note that the product here no longer includes a term for the rear-most layer. Similarly, eq. 4.16 becomes

$$P(m_i = k|\mathbf{h}_{1..K}^{(s)}, \pi) = \begin{cases} \displaystyle\prod_{k':\pi(k')<\pi(k)} \text{SHAPE}(s_{k',i} = 0|\mathbf{h}_k^{(s)}) & \text{if } k \text{ is rear-most} \\ \displaystyle\prod_{k':\pi(k')<\pi(k)} \text{SHAPE}(s_{k',i} = 0|\mathbf{h}_k'^{(s)}) & \\ \qquad \times \text{SHAPE}(s_{k,i} = 1|\mathbf{h}_k^{(s)}) & \text{otherwise.} \end{cases} \tag{4.21}$$

Note that if k is rear-most (i.e. if $\pi(k) = K$) then the term $\text{SHAPE}(s_{k,i} = 1|\mathbf{h}_k^{(s)})$ is missing; for all other cases, the proportionality $\propto$ in eq. 4.16 becomes an equality.

In our experiments (section 4.5) we have found that the scheme which replaces the sum over unobserved shape pixels with a sample from the posterior works well even when only a single sample is used. It should, however, be noted that the scheme does not implement an unbiased estimator of the expression in equation (4.17). To see this, we consider the general case of marginalizing with respect to some sub-set of variables

$$Z_s = \tilde{p}(\mathbf{s}_O) = \sum_{\mathbf{s}_U} \tilde{p}(\mathbf{s}_O, \mathbf{s}_U). \tag{4.22}$$

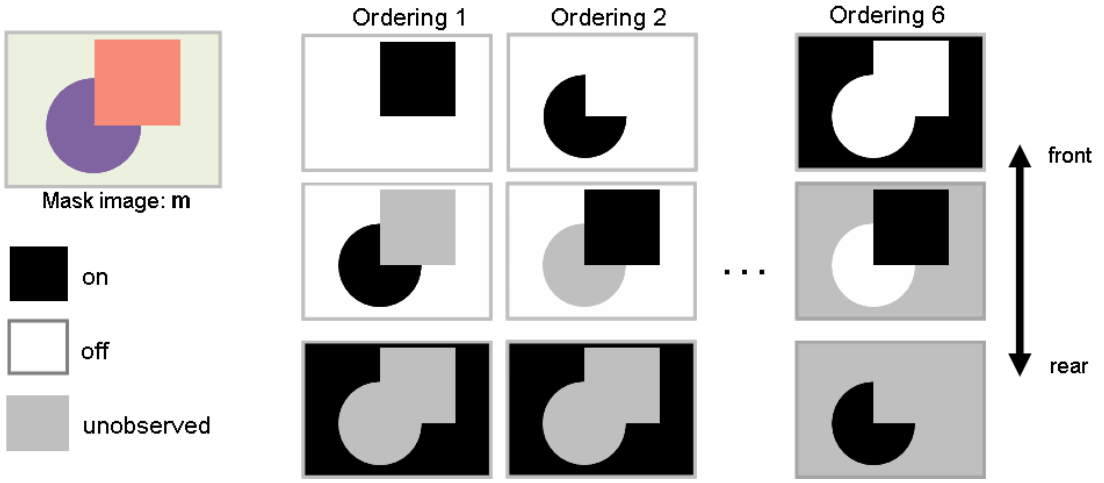


Figure 4.6: **Depth inference in the occlusion model.** The mask image (top left) comprises three regions, so there are $3! = 6$ possible depth orderings. Together with the mask, the ordering defines which shape pixels $\mathbf{s}_{k,i}$ are observed and which are unobserved. This is illustrated for three of the six possible orderings (white regions: shape off; black regions: shape on; gray regions: shape unobserved). Unobserved pixels (corresponding to $U_{\pi,k}(\mathbf{m})$ in eq. 4.17) can be "filled-in" by the shape model. Thus, for a shape model that favors circles, squares, and homogeneous backgrounds ordering 1 is preferable to all other orderings (including 2 and 6).

Here, we are using $\mathbf{s}_O$ for the set of observed variables and $\mathbf{s}_U$ for the unobserved variables; $\tilde{p}$ indicates the unnormalized distribution. This setup corresponds to the problem of computing a single factor of the right hand side of equation (4.17); $\mathbf{s}_U$ represents the unobserved shape pixels, and $\mathbf{s}_O$ the observed pixels. The approximation in (4.18) then corresponds to

$$\hat{Z}_s = \tilde{p}(\mathbf{s}_O, \hat{\mathbf{s}}_U), \tag{4.23}$$

where $\hat{\mathbf{s}}_U \sim p(\mathbf{s}_U|\mathbf{s}_O)$. In expectation this corresponds to computing

$$E\left[\hat{Z}_s\right] = \sum_{\mathbf{s}_U} p(\mathbf{s}_U|\mathbf{s}_O)\tilde{p}(\mathbf{s}_O, \mathbf{s}_U) \tag{4.24}$$

which is different from (4.22) and in general smaller than $\hat{Z}_s$ (since $\mathbf{s}_U$ is discrete and thus $p(\mathbf{s}_U|\mathbf{s}_O) \leqslant 1$). Furthermore, the bias will depend on the dimensionality of $\mathbf{s}_U$, and will be strongest when $p(\mathbf{s}_U|\mathbf{s}_O)$ is uniform but zero if $p(\mathbf{s}_U|\mathbf{s}_O) = \delta(\mathbf{s}_U = \mathbf{s}_U^0)$ for some $\mathbf{s}_U^0$. These considerations and simulations suggest that in equation (4.20) this bias is likely have the effect that depth orderings with fewer unobserved pixels

are preferred, although, as explained, the strength of this effect will depend on the uncertainty in $\text{SHAPE}(\mathbf{s}_{k,i:i\in U_{\pi,k}}|\mathbf{s}_{k,i:i\notin U_{\pi,k}})$, i.e. the uncertainty in the posterior over the unobserved pixels given the observed pixel, for all k. It is hard to assess the full effect of this bias in models of realistic size and in the context of full depth inference. In our experiments with the occlusion model with toy data it did not seem to have a detrimental effect (see section 4.5). Furthermore, it is possible to construct an estimator that is computationally still tractable but does not suffer from the same bias:

$$\hat{Z}'_s = \frac{\tilde{p}(\mathbf{s}_O, \hat{\mathbf{s}}_U)}{p(\hat{\mathbf{s}}_U|\hat{\mathbf{h}})} \tag{4.25}$$

where $\hat{\mathbf{s}}_U \sim p(\hat{\mathbf{s}}_U|\hat{\mathbf{h}})$; $\hat{\mathbf{h}} \sim p(\hat{\mathbf{h}}|\mathbf{s}_O)$ is obtained by running masked Gibbs sampling for a certain number of iterations. This estimator is very similar to (4.23) but takes into account the uncertainty in the posterior: It introduces a correction through dividing by the probability of our sample of the unobserved pixels $\hat{\mathbf{s}}_U$ under the conditional distribution from which it was sampled $p(\mathbf{s}_U|\mathbf{h})$. This can be thought of as an importance sampling estimator for which the proposal distribution ($p(\mathbf{s}_U|\mathbf{h})$ is constructed by running a Gibbs chain (cf. Neal (1993), equation 6.13). This estimator is unbiased

$$\begin{aligned}
E\left[\hat{Z}'_s\right] &= \sum_{\mathbf{s}_U,\mathbf{h}} p(\mathbf{s}_U, \mathbf{h}|\mathbf{s}_O) \frac{\tilde{p}(\mathbf{s}_O, \mathbf{s}_U)}{p(\mathbf{s}_U|\mathbf{h})} && (4.26)\\
&= \sum_{\mathbf{s}_U,\mathbf{h}} p(\mathbf{s}_U|\mathbf{h}) p(\mathbf{h}|\mathbf{s}_O) \frac{\tilde{p}(\mathbf{s}_O, \mathbf{s}_U)}{p(\mathbf{s}_U|\mathbf{h})} && (4.27)\\
&= \sum_{\mathbf{s}_U} \tilde{p}(\mathbf{s}_O, \mathbf{s}_U) \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{s}_O) && (4.28)\\
&= \tilde{p}(\mathbf{s}_O) && (4.29)
\end{aligned}$$

provided that we can obtain unbiased samples from $p(\hat{\mathbf{h}}|\mathbf{s}_O)$, i.e. provided that the Gibbs chains are run for long enough so that they reach equilibrium. We performed the experiments with the toy data described in section 4.5 (and also some other dataset) using both (4.25) and (4.23) but found that the results were largely identical, suggesting that the error introduced by (4.23) (and thus 4.20) is small relative to e.g. the variability that arises from using a sample-based approximation in the first place. All experimental results described below have been obtained with (4.23).

4.3.3.2 Learning

The parameters of the occlusion model that are to be learned are the parameters of the binary shape RBM $\Theta^{(s)} = (W^{(s)}, \mathbf{b}^{(s)}, \mathbf{c}^{(s)})$, i.e. the weight matrix and the biases

for the visible and hidden units. Training the occlusion model from a set of training mask images $\mathbf{m}^{(1)} \dots \mathbf{m}^{(T)}$ involves maximizing the likelihood of the model given the training data, i.e. finding Θ^* such that

$$\Theta^* = \text{argmax}_\Theta \sum_{t=1}^{T} \log P(\mathbf{m}^{(t)}) \tag{4.30}$$

$$= \text{argmax}_\Theta \sum_{t=1}^{T} \log \left[\sum_{\mathbf{s}_{1\dots K}, \mathbf{h}^{(s)}_{1\dots K}, \pi} P(\mathbf{m}^{(t)}, \mathbf{s}_{1\dots K}, \mathbf{h}^{(s)}_{1\dots K}, \pi) \right] \tag{4.31}$$

where $P(\mathbf{m}^{(t)}, \mathbf{s}_{1\dots K}, \mathbf{h}^{(s)}_{1\dots K}, \pi)$ is given by equation (4.12). Maximizing this likelihood is difficult for two reasons: Firstly, the sums over the unobserved variables $\mathbf{s}_{1\dots K}, \mathbf{h}^{(s)}_{1\dots K}, \pi$ cannot be computed. Secondly, $P(\mathbf{m}^{(t)}, \mathbf{s}_{1\dots K}, \mathbf{h}^{(s)}_{1\dots K}, \pi)$ involves the normalization constant of the shape RBM SHAPE$(\mathbf{s}, \mathbf{h})$ which is intractable as is the gradient of the normalization constant with respect to the model parameters. As discussed in chapter 2 above, the latter is a common problem encountered when training undirected graphical models including RBMs and alternative learning criteria (such as contrastive divergence (CD) learning proposed by Hinton, 2002) or stochastic maximum likelihood (Tieleman, 2008; Tieleman and Hinton, 2009) are therefore commonly used (see also discussion in section 2.3.2 of chapter 2) .

In our case, the situation is complicated by the fact that given a mask image $\mathbf{m}$ the latent shapes $\mathbf{s}_{1\dots K}$ are only partially observed. We therefore take the following approach: Given a set of training data points $\mathbf{m}^{(1)} \dots \mathbf{m}^{(T)}$ and current model parameters $\Theta^{(s)}$ we first perform inference with respect to the unobserved variables by drawing samples approximately from the posterior $P\left(\mathbf{s}^{(t)}_{1\dots K}, \pi^{(t)} | \mathbf{m}^{(t)}\right)$. (In practice, as discussed in section 2.1.2.2, we keep samples from the posterior from one iteration to the next and only perform a small number of Gibbs sampling steps in order to update this sample representation of the posterior.) We then use these sample latent shapes $\mathbf{s}^{(1)}_{1\dots K} \dots \mathbf{s}^{(T)}_{1\dots K}$ as "training data" to compute a contrastive divergence update step of the model parameters: For each mask image we first sample $\mathbf{h}^{(t)+}_{1\dots K}$ according to SHAPE$(\mathbf{h}^{(t)+}_k | \mathbf{s}^{(t)}_k)$ for the positive part of the gradient. We then obtain samples $\mathbf{s}^{(t)-}_{1\dots K}$, $\mathbf{h}^{(t)-}_{1\dots K}$ for computing the negative part of the gradient by performing block Gibbs sampling in the binary shape RBM, starting either at the data (for CD) or with the current particles in the persistent chains (for SML). For instance, in the case of CD-1 we first sample $\mathbf{s}^{(t)-}_{1\dots K} \sim$ SHAPE$(\cdot | \mathbf{h}^{(t)+}_k)$ and then $\mathbf{h}^{(t)-}_{1\dots K} \sim$ SHAPE$(\cdot | \mathbf{s}^{(t)-}_k)$. The gradient update

is then computed as follows:

$$\Delta\theta \propto -\frac{1}{T(K-1)}\sum_{t=1}^{T}\sum_{k:\pi_t(k)\neq K}\left(\frac{\partial}{\partial\theta}E_s(\mathbf{s}_k^{(t)},h_k^{(t)^+})-\frac{\partial}{\partial\theta}E_s(s_k^{(t)-},h_k^{(t)-})\right), \tag{4.32}$$

where E_s is the energy of the binary shape RBM and θ a parameter of the energy function. Note that (4.32) is similar to the update for the softmax model described above (equation 4.11). There is, however, one important difference: in order to collect statistics for the negative part of the gradient there is no need to actually sample a mask image from the full occlusion model. Due to the marginal independence of the $\mathbf{s}_1 \ldots \mathbf{s}_K$ it is sufficient to draw samples from the shape RBM without having to compose them to an actual mask image $\mathbf{m}$. Note further that the rear-most shape $\mathbf{s}_{\pi^{-1}(K)}$ is not included in the update.

4.3.4 Integrating the mask prior with the masked RBM

Both the softmax model and the occlusion model can be integrated with the masked RBM in a straightforward manner (integrating the uniform shape model is trivial, there is nothing to do). Key is again the fact that for both models the conditional distributions over the mask pixels are factorial given the state of the hidden units and, for the occlusion model, the depth ordering (cf. equations (4.7) and (4.16)). The contribution of the appearances to the conditional probability over the mask is readily combined with the contribution from the shapes. For the softmax model we obtain:

$$P(\mathbf{m}|\mathbf{v},\mathbf{h}^{(a)}_{1\ldots K},\mathbf{h}^{(a)}_{1\ldots K}) = \prod_i P(m_i|v_i,\mathbf{h}^{(a)}_{1\ldots K},\mathbf{h}^{(s)}_{1\ldots K}) \tag{4.33}$$

$$P(m_i=k|v_i,\mathbf{h}^{(a)}_{1\ldots K},\mathbf{h}^{(s)}_{1\ldots K}) \propto \mathrm{SHAPE}(s_{k,i}=1|\mathbf{h}^{(s)}_k)\mathrm{APP}(v_i|\mathbf{h}^{(a)}_k) \tag{4.34}$$

Similarly, for the occlusion model we find

$$P(\mathbf{m}|\mathbf{v},\mathbf{h}^{(a)}_{1\ldots K},\mathbf{h}^{(s)}_{1\ldots K},\pi) = \prod_i P(m_i|v_i,\mathbf{h}^{(a)}_{1\ldots K},\mathbf{h}^{(s)}_{1\ldots K},\pi) \tag{4.35}$$

$$\begin{aligned} P(m_i=k|v_i,\mathbf{h}^{(a)}_{1\ldots K},\mathbf{h}^{(s)}_{1\ldots K},\pi) \propto\ & \mathrm{APP}(v_i|\mathbf{h}^{(a)}_k)\mathrm{SHAPE}(s_{k,i}=1|\mathbf{h}^{(s)}_k) \\ & \times \prod_{k':\pi(k')<\pi(k)}\left[1-\mathrm{SHAPE}(s_{k',i}=1|\mathbf{h}'^{(s)}_k)\right] \end{aligned} \tag{4.36}$$

This suggests a Gibbs sampling scheme in which we perform inference in the full model alternating the following two steps:

1. Given a mask, we update the unobserved variables in the shape and appearance parts of the model, in particular the shape and appearance hidden units $\mathbf{h}^{(a)}_{1\ldots K}$ and

$\mathbf{h}^{(s)}_{1...K}$, as well as the unobserved shape pixels and the depth ordering π for the occlusion model.

2. Given the shape and appearance hidden units $\mathbf{h}^{(a)}_{1...K}$ and $\mathbf{h}^{(s)}_{1...K}$ (and the depth ordering π in the case of the occlusion model) we update the mask using equations (4.34) or (4.36).

Note that step 1) can be performed independently for the shape and appearance parts of the model. The scheme can be initialized using a random mask and a random assignment of the unobserved pixels of latent appearances (and shapes for the occlusion model) and it allows us to perform global inference given an image. It also allows to perform joint training of the shape and appearance model, although in most experiments described in the remainder of this chapter the two parts of the model are trained primarily separately. The appearance model is usually learned first, a shape model is then learned using this preliminary appearance model, finally both models can be fine-tuned together by performing joint learning. For details see section 4.5 below.

4.4 Related work

In the previous section we have described a model of natural images that is based on concepts from the deep learning literature (in particular Restricted Boltzmann Machines, RBMs, e.g. Smolensky, 1986; Hinton et al., 2006b) and that explicitly incorporates some knowledge about how images are formed. One of the motivations for the model was that standard RBMs, such as the Gauss-Bernoulli-RBM (cf. section 2.2.4, and e.g. Hinton and Salakhutdinov, 2006; Lee et al., 2009) are relatively inefficient models of generic natural image structure. In the full model defined in the previous section, an image is composed from several independent, potentially overlapping and *occluding* objects. Also, the model attempts to efficiently capture some of the variability observed in natural images by factorizing shape and appearance.

In this review we will focus on two lines of work. Section 4.4.1 will discuss the large body of previous work that has attempted to define generative models that capture some of the statistical properties of natural images. For computational reasons these models typically focus on image patches (as will our experiments described in section 4.5). Related models that have been formulated for larger images will be discussed in the section 5.2 of the next chapter. Many of the models that we will discuss in section 4.4.1 share the underlying belief that an image should be explainable in terms

of a relatively small number of independent causes. An important feature of many of these models is further that they define the interaction between causes in terms of a linear superposition of basis functions which is in stark contrast to the highly nonlinear interaction arising from the occlusion operation described above. Although there have also been attempts to define models with non-linear interactions, most of these models still fall short of properly modeling the occlusion non-linearity, and they have typically been applied only to rather limited datasets.

A second line of work that will be discussed in section 4.4.2 has focused on learning layered representations of a small set of homogeneous images, such as the frames of a movie sequence. While these models explicitly reason about occlusions and the depth ordering of objects in an image, they are usually limited to a small number of specific objects that occur in the image set and which are described in terms of shape and appearance templates.

4.4.1 Generative models of natural image statistics

A large body of literature has been devoted toward modeling the statistical properties of natural images. This problem has achieved great attention in the computer vision as well as in the neuroscience literature.

Two related models that are more or less inextricably linked with the idea of modeling image statistics are the sparse coding model suggested by Olshausen and Field (1997) and the work on independent component analysis (ICA) by Bell and Sejnowski (1997). These models have motivated a large body of follow-up work that extend the basic ideas in various ways. Although the two models differ in their details, they share two important ideas: independence and sparsity. As explained in section 2.2.2 of chapter 2 this means that any given image should be explainable in terms of a relatively small number (out of possibly a large dictionary) of independent causes. In practice this is often implemented in terms of a sparse, independent prior over latent variables that control the activation of basis functions that interact linearly to form the image (cf. equations (2.12, 2.14) in section 2.2.2). The parameters of these models can be fully learned from data.

Despite the popularity of this modeling approach it is generally acknowledged that the assumptions underlying these models do not capture the image formation process well. Various authors have, noted, for instance, that the "independent components" discovered by these approaches are in fact far from independent (e.g. Buccigrossi and

Simoncelli, 1999; Bethge, 2006). One potential reason for this is that while the notion of independent causes (which are often related to the objects that comprise an image) is conceptually appealing, there seems to be a significant gap between this concept and its implementation in terms of basis functions that interact linearly. As has already been acknowledged e.g. in Olshausen and Field (1997) such a linear interaction is a very crude approximation to, for example, the highly non-linear interaction when objects occlude each other. This has been the motivation for several models that use alternative operations to model the interactions of causes: Examples can be found, for instance, in Saund (1995); Dayan and Zemel (1995); Lee and Seung (1999); Ross and Zemel (2006); Lücke and Sahani (2008); Lücke et al. (2009); Puertas et al. (2010), although some of these models have been investigated only for rather simple scenarios and not for modeling natural images (but see Lücke and Sahani, 2008; Puertas et al., 2010). The approach proposed in Lücke et al. (2009) is noteworthy in that it models occlusion explicitly. However, in the form that is presented it is currently limited to a small dictionary of objects with simple appearances (an object is represented by a shape template and single color) and not suitable for modeling natural image patches. The model proposed by Ross and Zemel (2006) bears some similarity to the masked RBM in that it also decomposes an image into several regions that are goverend by independent appearances using a mask, but mask and appearance model are much simpler (in particular there is no notion of occluding shapes and the mask model is simply a pixel-wise independent multinomial distribution) and the model is applied to decompose sets of homogeneous images (e.g. of faces) into qualitatively different parts.

Another set of models have been devised to relax the strong independence assumptions imposed by sparse coding and ICA (e.g. Hyvärinen and Hoyer, 2000; Hyvärinen et al., 2001; Sinz and Bethge, 2009; Sinz et al., 2010; Karklin and Lewicki, 2009). One prominent example of this work is, for instance, subspace ICA proposed by Hyvärinen and Hoyer (2000) which instead of maximizing the independence of the responses of linear filters, attempts to maximize the independence between the norms of projections on linear subspaces. More recently Karklin and Lewicki (2009) have proposed a model in which the likelihood is Gaussian as in the model proposed by Olshausen and Field (1997) (cf. equations (2.12, 2.14)), but in which the latent variables directly control the conditional covariance of the likelihood instead of just the mean. The intuitive motivation is that different configurations of the latent variables represent the statistical variations that characterize local image regions and thus provide a more abstract repre-

sentation of the local image characteristics than for instance in the model by Olshausen and Field (1997).

The models discussed above are all causal (i.e. directed models). As discussed in chapter 2 (section 2.2.3) undirected, Product-of-Experts (PoE) models have also been applied to the problem of modeling natural image patches. For instance, Teh et al. (2003) propose a PoE model that can be seen as the undirected equivalent of the sparse coding models discussed above, and is equivalent to ICA in the complete case. The hierarchical PoE model described in Osindero et al. (2006) bears similarity to the topographic ICA model proposed in Hyvärinen et al. (2001). Very recently Ranzato et al. (2010a); Ranzato and Hinton (2010) have proposed a RBM, the mcRBM, in which the conditional distribution over the visibles given the hidden $p(\mathbf{v}|\mathbf{h})$ is a Gaussian with a *covariance* matrix that depends on $\mathbf{h}$ (thus being closely related to the work by Karklin and Lewicki (2009) discussed above). This is contrast to previous work on continuous-valued RBMs for which the distribution over the visible units was also conditionally Gaussian but for which only the mean depended on the state of the hidden units while the variance was fixed and the visible units conditionally independent. There is an interesting connection between the masked RBM and the models that explicitly model the covariance structure. In fact, the model in Ranzato et al. (2010a); Ranzato and Hinton (2010) has been motivated in a manner very similar to the masked RBM, i.e. in terms of edges and in terms of the break-down of correlations across edges. In the masked RBM the mask partitions the image pixels into different regions and pixels within each region covary as prescribed by the RBM used to model the appearances. In the covariance RBM (cRBM; Ranzato et al., 2010a) and the mean-covariance RBM (mcRBM; Ranzato and Hinton, 2010) a similar *soft* partitioning can be achieved by choosing a configuration of the hidden units that leads to a covariance matrix that introduces strong correlations between visible units belonging to the same region but no correlations between pixels in different regions. Unlike in the masked RBM, however, in which the mask always leads to a hard partitioning into fully independent region this is unlikely to arise in the mcRBM. Furthermore, in the mcRBM there is no explicit notion of shape or of occlusion.

4.4.2 Modeling shape and appearance of objects

The work described in the previous section is concerned with generative models of general natural images. A rather different line of work has attempted to model small sets of

related images such as the frames of a movies or images of objects of a particular category. Layered representations that represent an image in terms of several overlapping objects which are arranged according to a depth ordering have been suggested at least 20 years ago (Adelson, 1991; Wang and Adelson, 1994) and have, since then been extensively used and formulated as probabilistic, generative models (e.g. Jojic and Frey, 2001; Frey and Jojic, 2003; Williams and Titsias, 2004; Titsias and Williams, 2004; Kannan et al., 2005; Winn and Jojic, 2005). Similar to our masked RBM with occlusion shape model these approaches represent an image in terms of several objects with associated shapes and appearances. The shape is typically a binary image as in the occlusion model for the masked RBM (but see Jojic and Frey, 2001; Frey and Jojic, 2003 who allow for a real-valued mask that can account for blending). Shape and appearance are then combined to form an image according to a relative depth ordering. These approaches are more general than the masked RBM in the sense that they also explicitly model transformations (such as translation, rotation, or scaling) of the individual objects as well as local deformations. Global transformations are typically modeled by applying an appropriate transformation matrix to both the binary shape image and the appearance (there is a separate transformation matrix for each possible transformation, e.g. each possible shift). Kannan et al. (2005); Winn and Jojic (2005) model local deformations by estimating a deformation field that comprises a deformation vector for each pixel. For instance, the generative model of Williams and Titsias (2004) for an image $\mathbf{x}$ containing two objects and a background is given by

$$\begin{aligned} p(\mathbf{x}|t_1,t_2,t_B) &= \prod_i \Big[\big([T_{t_1}\boldsymbol{\pi}_1]_i \mathcal{N}(x_i;[T_{t_1}\mathbf{f}_1]_i,\sigma_1^2) \big) \\ &\quad + (1-[T_{t_1}\boldsymbol{\pi}_1]_i) \big([T_{t_2}\boldsymbol{\pi}_2]_i \mathcal{N}(x_i;[T_{t_2}\mathbf{f}_2]_i,\sigma_2^2) \\ &\quad + (1-[T_{t_2}\boldsymbol{\pi}]_i) \mathcal{N}(x_i;[T_{t_B}\mathbf{b}]_i,\sigma_B^2) \big) \Big]. \end{aligned} \tag{4.37}$$

The two foreground objects are indicated by the subscripts $_1$, $_2$, the background by $_B$. The shapes of foreground objects are modeled using multivariate (pixel-wise independent) Bernoulli distributions (pixel i is part of object k with probability $\pi_{k,i}$; the background is assumed to be present everywhere and its shape is therefore not represented explicitly). The appearances are modeled in terms of the mean color for each pixel that is part of the object ($f_{k,i}$ indicates the mean color of pixel i of object k for the foreground objects; b_i is the same for the background). T_t represent transformation matrices which are applied to both shapes and appearances and can realize, for instance, translations or rotations (t_1 ,t_2, and t_B index the selected transformation for the two foreground objects and the background respectively).

Whereas the handling of occlusions in equation (4.37) is very similar to the masked RBM, the scope of these models is rather different: They are typically fitted to sequences of images with a small number of specific objects (such as a background and two foreground objects; cf. Fig. 4.7 for a typical example). The depth ordering of these objects is fixed and individual objects are represented either in terms of fixed templates (as in equation 4.37) or in terms of mixtures of such templates with a small number of components or as linear manifolds. This is rather different from the masked RBM with occlusion shape model which treats all regions (objects) as being equal (i.e. all objects share the same appearance and shape model; note that there are different sets of parameters for the different objects in equation 4.37). Furthermore the masked RBM with occlusion shape model does not explicitly model transformations or deformations, but uses very rich priors (RBMs) to model shapes and appearances of objects that can appear in the dataset.

A further difference between these layered image representation and the masked RBM is the way inference is performed. In general, inference in layered image models is very expensive since it effectively involves a combinatorial search over all depth orderings and all possible transformations / deformations of all objects. Even for moderately sized images this number is very large so that the exact posterior is not tractable. Two general strategies have been developed to overcome this problem: Jojic and Frey (2001); Frey and Jojic (2003); Kannan et al. (2005) use various variational approaches typically involving factorized approximate posterior distributions (see Frey and Jojic (2005) for a review) while Williams and Titsias (2004); Titsias and Williams (2004) develop a very efficient approach that performs inference (and learning) greedily, one object at a time way thus avoiding the combinatorial explosion otherwise encountered. Both approaches are rather different from the sampling-based inference scheme described in sections 4.3.3 and 4.3.4 above.

4.5 Experiments

The experiments in this section provide an evaluation of the occlusion based shape model on toy data and real data. They focus on the following questions:

- Is the formulation of the occlusion shape model viable? Can inference and learning be performed as described above?
- Is there an advantage of the occlusion shape model over the simpler softmax

Figure 4.7: **Typical training data for classical layered image models**: Four frames of a representative video sequence that has been modeled using the layered representations discussed in section 4.4.2. (Figure taken from Williams and Titsias (2004))

model?

- Does the masked RBM with the occlusion based shape model give rise to a good generative model of natural images?

The masked RBM is a partially undirected graphical model. The evaluation of undirected graphical models is notoriously difficult since normalization constants are typically not tractable and a computation of the likelihood of data under such a model, the most natural way of assessing a model's quality, is therefore usually not possible. Surrogate measures are therefore typically used (see also discussion in chapter 2), and we follow the same approach here.

In section 4.5.1 we will first evaluate the occlusion shape model for the mask described in section 4.3.2 in isolation and compare it to the softmax model (section 4.3.1). For this purpose, we will use as training data a toy dataset of K-valued *mask* images. The purpose of this evaluation is to demonstrate the general viability of the occlusion model and also its superiority over the softmax model. Working with mask images directly decouples the problem of learning a prior over masks (i.e. a prior over segmentations) from the problem of actually inferring the correct segmentation from a RGB image. It therefore makes it possible to directly assess the ability of the model to learn about shapes and to deal with occlusion. (In the full masked RBM the inferred segmentation is determined by the model of the mask as well as by the appearance model.) Using a toy dataset further has the advantage that we have ground truth avail-

able regarding the shapes that we expect the model to learn and also with respect to correct depth ordering of shapes in an image.

In section 4.5.2 we evaluate the ability of the masked RBM with occlusion shape model to learn generative models of natural image patches. These experiments serve (a) to demonstrate the viability of the occlusion shape model in the context of the masked RBM, (b) to demonstrate that the occlusion shape model is able to learn sensible shape priors even for complex scenarios, and (c) to demonstrate that the factorization of shape and appearance is an efficient way of representing the structure in natural image patches.

4.5.1 The benefits of modeling occlusion explicitly: softmax vs. occlusion

This section presents an evaluation of the occlusion shape model on simple toy data. These experiments serve to demonstrate the general viability of the occlusion model and also its superiority over the softmax model. In particular, we aim to demonstrate that (a) inference and learning is indeed feasible in the occlusion model, (b) the representation learned by the occlusion model is more efficient than the representation acquired with a softmax model, and (c) explicit knowledge about occlusions leads to more robust performance in a recognition task.

4.5.1.1 Methods & Dataset

Dataset: The toy masks dataset is composed of 4000 14×14 mask patches (a mask patch is a 3-valued image, i.e. each pixel takes values $m_i \in \{1,2,3\}$) generated from the superposition of an MNIST digit (from the class "3") and a shape (a circle, a square or a triangle). In this dataset, neither digits nor shapes are shown in isolation, and each digit example appears only in exactly one image. Since the digit is in the background in half of the patches, half of the digit examples are only partially visible. Example masks from this dataset are shown in the top panel of Fig. 4.8: each pixel can take three values (represented by different gray levels), one for each object in the patch (the background being the third object). Which value is used to represent each object is irrelevant; the actual values are not used to infer the depth ordering.

Models & Learning: We trained mask models with three layers ($K = 3$). We trained one occlusion-based mask model with 20 hidden units, and, for comparison, two softmax models with 20 and 70 hidden units respectively.

The softmax models were trained with stochastic maximum likelihood ("persistent CD", Tieleman (2008); cf. section 2.3.2), a learning rate of 0.01, momentum of 0.5, weight decay 0.0005. The size of our mini batches was 100, and we used 200 particles for the persistent chains. Training was performed for 5000 epochs (each epoch corresponding to one full sweep through the dataset). The occlusion model was also trained with stochastic approximation, using the same learning parameters and batch size.

Training the occlusion model requires some care at the beginning of learning. As explained in section 4.3.3.2 learning requires inference with respect to the depth ordering and the latent shapes. Very early in learning (after a random initialization of the weights) the model has a tendency to place large shapes in the foreground allowing it to hallucinate arbitrary shapes in the largely occluded layers, and as a consequence the model parameters tend to quickly converge toward a degenerate local minimum from which learning will not recover. There are various ways to overcome this problem. For the experiments with toy data presented here, we used a short "annealing" phase at the beginning of learning: During this phase we applied a temperature $T > 1$ when sampling the depth ordering π. Annealing was used for the first 20 mini batches and the temperature was decreased from initially 20 to 1.

4.5.1.2 Results

Fig. 4.8 shows samples from the occlusion model with 20 hidden units, the softmax model with 70 hidden units, and the softmax model with 20 hidden units. Samples from the occlusion model are obtained by drawing two independent samples from the shape RBM for the top-most and second-most layer and then composing these samples as prescribed by eq. 4.12 to generate the full image. The softmax model with 70 hidden units per layer generates good samples. Yet, when limiting the capacity and using only 20 hidden units as for the occlusion model, the samples drawn from the occlusion based mask model are considerably more convincing than those drawn from the softmax model. Indeed, the latter generated samples with improper occlusions or deformed digits. It is also interesting to note that the occlusion model generalized to samples not seen in the training set, like the two MNIST digits occluding each other which is due to the fact that the shapes in the different layers are sampled independently of each other (one might debate whether this is a desirable effect or not).

Figure 4.9 gives some insight into the latent representation of the two models. The left hand panel shows a subset of the samples from the occlusion model together with the corresponding two independent samples from the shape RBM that have been

superimposed to generate the full sample. This demonstrates that the occlusion model has indeed learned a model of the individual shapes despite the fact that it has never seen them in isolation. In the softmax model, on the other hand, the layers cooperate to generate a particular image of occluding shapes. It is not possible to sample from the individual layers separately, but one can still inspect the inputs to the three layers of visible units which are tied together by the softmax (cf. equation 4.10; the input to a given unit $s_{k,i}$ is given by $W_{i\cdot}^{(s)}\mathbf{h}_k^{(s)} + b_i^{(s)}$). These inputs are shown in Fig. 4.9 in the right-hand panel. It is clear that no shape is generated by a single layer but that all three layers have to interact. In the first row, for instance, all three inputs contain a "3" (either with positive or negative weights). These cancellations are inevitable for the softmax model to generate confident samples. While the occlusion model learns about the individual image elements, the softmax model has to represent all their possible arrangements explicitly, which is less efficient and thus requires a larger number of hidden units. This also leads to a set of hidden units which is far less indicative of the shape in the image than in the occlusion model as we will demonstrate in the next section.

4.5.1.3 Sensitivity to occlusion

To assess the importance of the difference in representation between the softmax and the occlusion mask models in a recognition task, we created pairs of images containing one digit and one shape. The same digit and the same shape were used in both images: In the first image, the digit was in front of the shape; in the second image, the digit was occluded by the shape (see inset of Fig. 4.10a for an illustration). For each image pair we then inferred the latent representations (the state of the hidden units) of the digit in the occluded and in the non-occluded condition and computed the root mean squared difference between the corresponding representations. As our main motivation is to recognize objects whether or not they are occluded, we would like the shape hidden units to be as similar as possible in the two cases. The occlusion based mask model clearly outperforms the softmax model (with 20 hidden units per layer), as may be seen in Fig. 4.10. Furthermore, in our experiments, the occlusion model inferred the correct ordering more than 95% of the time (chance being 17%, as there are three layers and thus six possible orderings).

In summary this toy dataset emphasizes the need for modeling occlusion when extracting a meaningful representation of the shapes present in images.

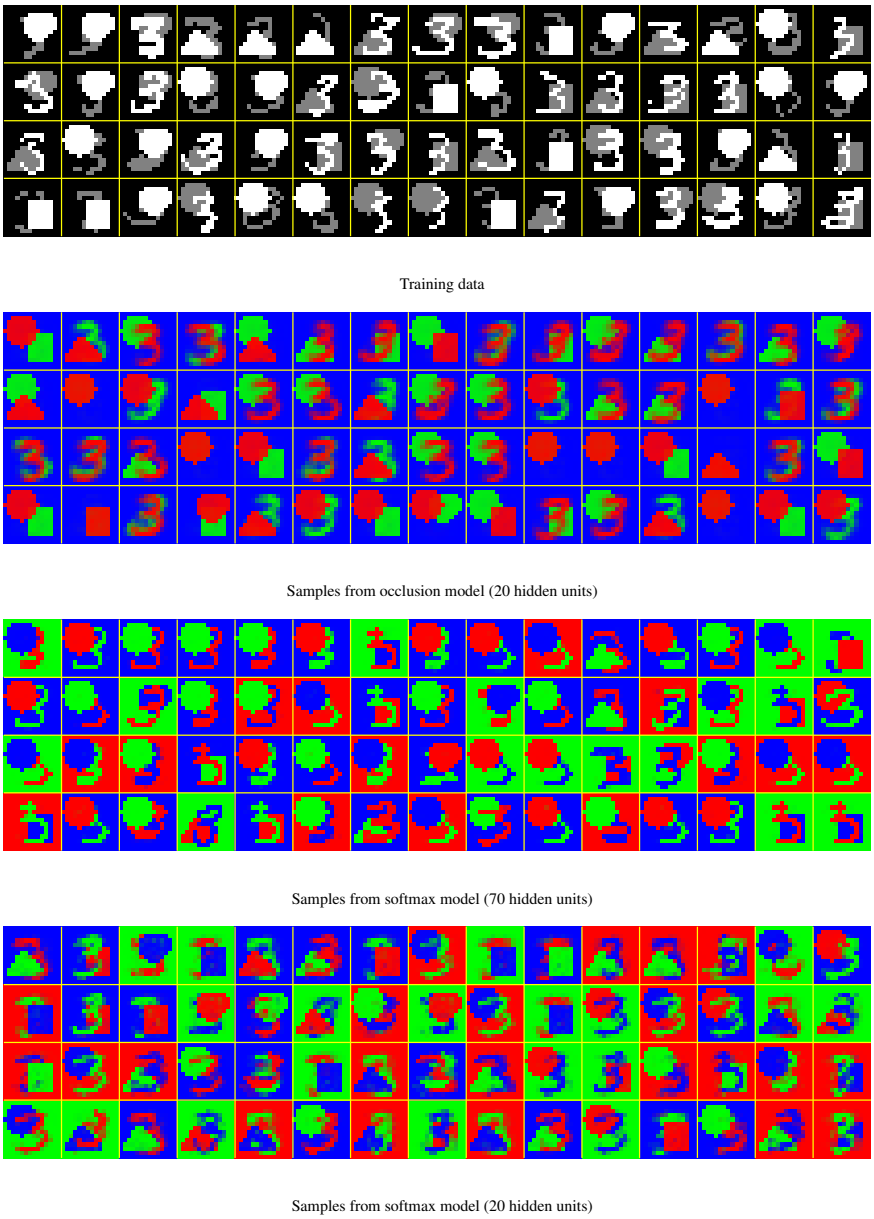


Figure 4.8: **Learning shapes under occlusion**: Training data and samples from the three learned models: occlusion model with 20 hidden units per layer, softmax model with 70 hidden units per layer, and softmax model with 20 hidden units per layer. The softmax model's performance decreases when it has limited capacity, yielding unconfident / invalid samples in the bottom panel. The independence between the layers allows the occlusion model to "generalize" to shape configurations it has not seen before: It generates samples containing only digits or only shapes. Samples that appear to contain only one shape arise from the model generating the same shape in both layers.

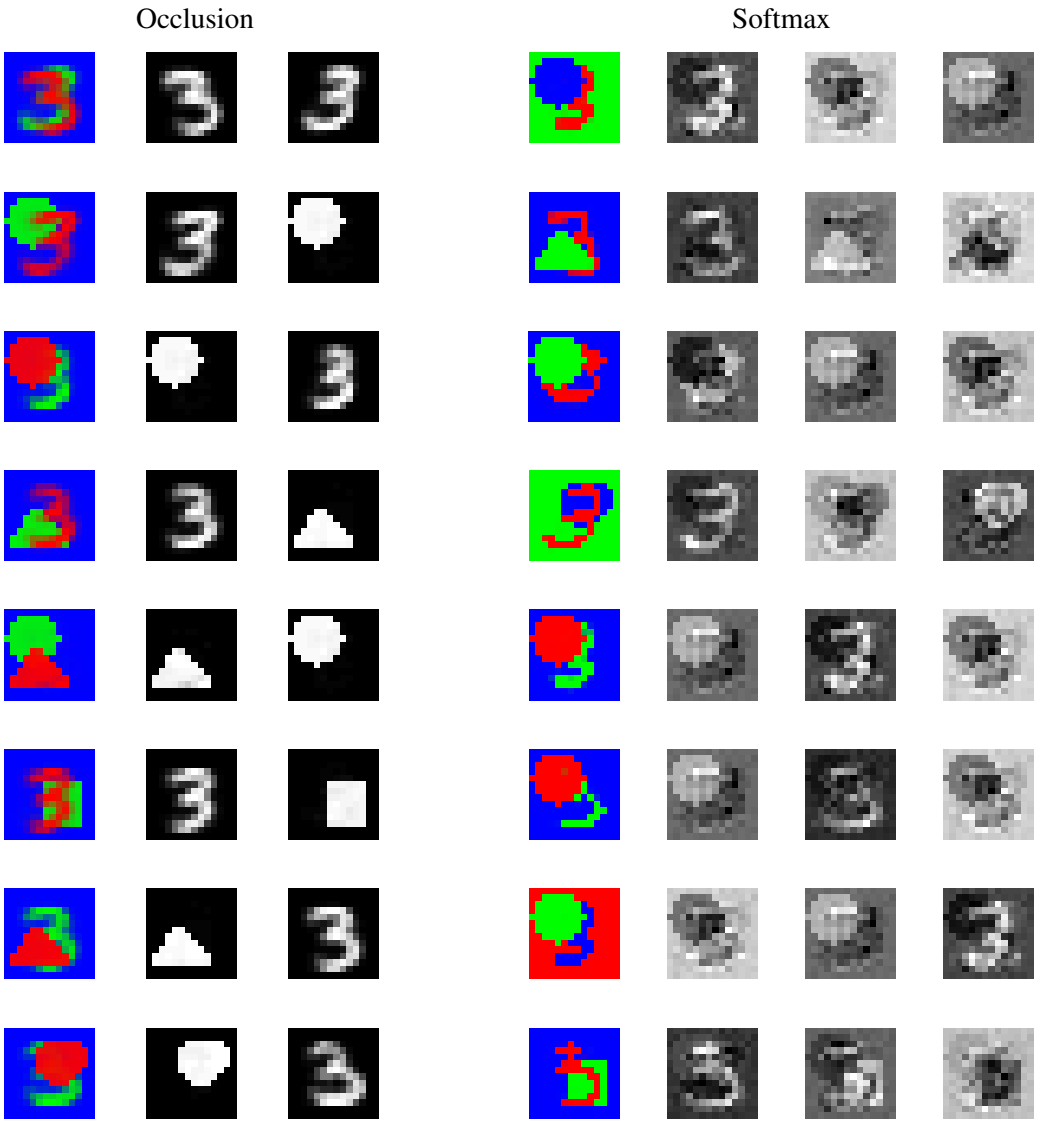


Figure 4.9: **Shape representation in the occlusion and softmax model**: Full samples and latent representation for the occlusion (*left*) and softmax model with 70 hidden units (*right*). The left panel shows for the occlusion model a subset of full mask samples together with the two independent samples from the shape RBM that have been composed to generate the full sample. For the softmax model the right panel shows the full samples together with the corresponding inputs to the softmax-function from the three sets of hidden units / layers (cf. equation 4.10). Note how the three layers interact to generate confident samples.

4.5.2 Modeling natural image patches

The experiments on toy data demonstrated that the occlusion model is able to learn and recognize shapes under occlusion and is able to perform depth inference given a

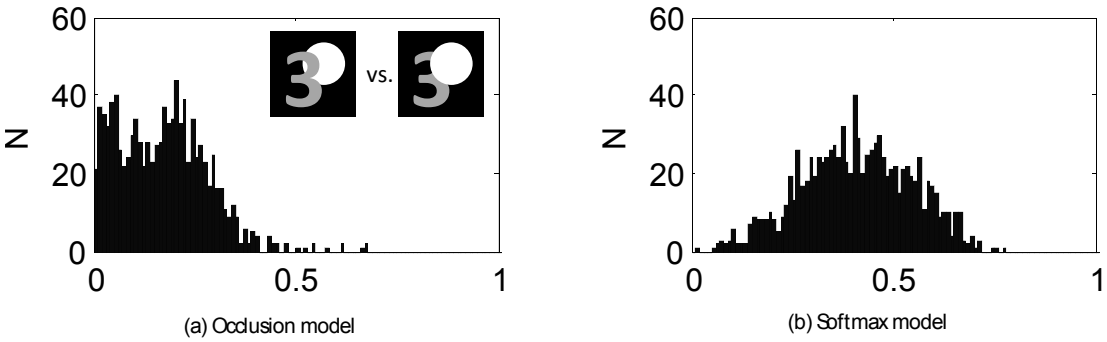


Figure 4.10: Shape recognition under occlusion: Histograms of the root mean squared differences between the latent representations (activation of the hidden units) of digits inferred when the digit is shown in the foreground and when it is occluded (see inset in panel *(a)* for an illustration of the two conditions). Results are shown *(a)* for the occlusion model and *(b)* for the softmax model (both with 20 hidden units per layer). Each datapoint in the histograms corresponds to one pair of images showing a digit in front and behind a shape.

mask image with occluding shapes. The second set of experiments on natural images assesses the joint model consisting of the shape and the appearance model. The experiment serves to demonstrate that the model can learn about shapes and reason about relative depth even when the dataset is complex and heterogeneous. It further demonstrates that the masked RBM endowed with a suitable shape model can give rise to a good model of the mid-level structure in natural image patches.

4.5.2.1 Methods & Dataset

Dataset: The dataset consisted of 21000 16×16 patches extracted from natural color images. Color patches were extracted randomly from images from the PASCAL VOC 2009 dataset[5]. No pre-processing was applied except that all images were scaled to be 320 pixels wide (maintaining their aspect ratio) prior to extracting the patches. Some example patches from the training data are shown in Figure 4.11.

Appearance RBM: The experiments with natural image patches require a continuous valued RBM as appearance model to model the RGB values of the image pixels. For the experiments described in this section we used the Beta RBM proposed in Le Roux et al. (2011); the energy function of this RBM is given in equation (4.38). Note that this formulation differs from the Beta RBM described in Welling et al. (2004)

[5]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2009/index.html

in that in (4.38) each expert is a mixture of two Beta distributions instead of a mixture between a Beta distribution and a uniform distribution. This symmetrizes the hidden units and allows for weaker constraints on the parameters while still retaining valid distributions.

$$\begin{aligned} E(\mathbf{v},\mathbf{h}) &= -\log(\mathbf{v})^T W^1 \mathbf{h} - \log(\mathbf{v})^T W^2 (\mathbf{e}-\mathbf{h}) \\ &\quad -\log(\mathbf{e}-\mathbf{v})^T U^1 \mathbf{h} - \log(1-\mathbf{v})^T U^2 (\mathbf{e}-\mathbf{h}) \\ &\quad +\mathbf{e}^T \log(\mathbf{v}) + \mathbf{e}^T \log(\mathbf{e}-\mathbf{v}) - \mathbf{c}^T \mathbf{h} \,. \end{aligned} \tag{4.38}$$

The elements of W_1, W_2, U_1, and U_2 are restricted to be positive. There are no visible biases since these can be absorbed into the weight matrices. The conditional distribution over the visible units is given by $p(\mathbf{v}|\mathbf{h}) = \prod_i p(v_i|\mathbf{h})$ where $p(v_i|\mathbf{h}) = \text{Beta}(v_i|\alpha_i(\mathbf{h}),\beta_i(\mathbf{h}))$ with $\alpha_i(\mathbf{h}) = W^1_{i\cdot}\mathbf{h} + W^2_{i\cdot}(\mathbf{1}-\mathbf{h})$ and $\beta_i(\mathbf{h}) = U^1_{i\cdot}\mathbf{h} + U^2_{i\cdot}(\mathbf{1}-\mathbf{h})$. In the experiments we used a Beta RBM with 128 hidden units. This RBM was pre-trained on 16×16 patches extracted from natural color images as described in Le Roux et al. (2011).

Other choices for the appearance RBM are possible. One conceivable alternative is, for instance, a simple Gaussian RBM with fixed variance as used e.g. in Hinton and Salakhutdinov (2006); Lee et al. (2009) (see also discussion in chapter 2, section 2.2.4). Unlike the Beta RBM, however, the latter does not model the local variance which is important for obtaining confident segmentations (note that the conditional distribution over the mask in equations (4.36) and (4.34) involve the probability of a pixel given the states of the hidden units associated with the different layers; the log-probability differences depend on the conditional variances as well as the means).

Learning: We trained an occlusion shape model with 384 hidden units in the context of the full masked RBM with K=3 on the data set described above. Training proceeded in two phases:

1. In the first phase we pre-trained the shape model directly on binary mask patches. For this purpose we inferred the mask ($K = 3$) for a large set of natural image patches (16×16 pixels RGB patches) using the *uniform* model as the mask prior. For each patch we performed 100 mask iterations to infer the mask with up to $K = 3$ regions (due to the lack of a shape prior, many of these masks were very noisy). From each 3-valued mask patch we obtained three binary patches, one for each region of the mask, and then trained a binary RBM (384 hidden units, 256 visible units) directly on 95000 of these binary patches. Training was performed with stochastic maximum likelihood (Tieleman and Hinton, 2009) with

a small learning rate of 0.0005, weight decay 0.0002, no momentum and mini-batches of size 100. Training was performed for 10000 epochs. The parameters of this binary RBM served as initialization for training of the shape model in the context of the full model. Pre-training took about 3.5 days using our Matlab implementation on a single-core machine.

2. In the second phase we trained the shape model in the context of the full model (masked RBM with $K = 3$). The parameters of the shape RBM were initialized with the parameters obtained from phase 1. We used a training set of 21000 RGB patches grouped into mini batches of size 60. Learning was performed in alternation with inference. For each patch we performed two iterations of full inference in the model (this includes the update of the appearance fantasies, of the depth, of the shape fantasies, and of the mask) before updating the model parameters. Inference was performed as described in section 4.3.3. During inference in the mask model we used 10 iterations of masked Gibbs sampling to update the shape fantasies. Before sampling, unobserved pixels in the shape fantasies were initialized with their state from the previous cycle. To prevent the model from hallucinating shapes into unused layers (which would slow down learning) we forced such layers to be in front of all visible layers and thus to be empty (all visible units off). Learning was performed using CD-10 with a learning rate of 0.001, weight decay of 0.0002 and a momentum of 0.5. Training in the full model was performed for 550 epochs and took approximately two weeks using our unoptimized Matlab implementation on a single-core machine.

Sampling from the model: Generating samples from the full model requires sampling from the appearance and the shape model. Sampling from the appearance model is difficult as its Gibbs chains tend to mix very poorly. This is due to the fact that the Beta RBM models not only the mean but also the variance of the visible units, which allows the conditional distributions to become very peaked. We therefore trained a second-layer (binary) RBM for the appearance model (i.e. turning it into a Deep Belief Network). This allowed us to generate samples from the appearance model by running Gibbs chains for 5000 steps of Gibbs sampling in the second layer RBM (which is a binary RBM and thus mixes more readily than the Beta RBM) to obtain samples of $\mathbf{h}^{(a)}$. Samples of $\hat{\mathbf{v}}_k$ were then generated by sampling from $\text{APP}(\hat{\mathbf{v}}|\mathbf{h}^{(a)})$ defined by the Beta RBM. More details of this scheme can be found in Le Roux et al. (2011). Samples from the shape model were obtained by running Gibbs chains for 15000 steps in

the binary shape RBM. As explained in section 4.3.2 we generate full shape samples by combining $K = 3$ IID samples from the appearance model with 2 IID samples from the shape model according to a random depth ordering to obtain a full sample from the masked RBM. The DBN for the appearance model was trained and the samples from the appearance were generated by Nicolas Le Roux.

Evaluation: Since computing the log-likelihood of test data under our model is not possible we evaluated it on two surrogate tasks. Firstly, we generated samples from the full model and compared them to the training data using visual comparison. One hallmark of natural images are the highly kurtotic filter response marginals. To further assess the quality of our sample patches we therefore also computed the response marginal of a set of Gabor and random filters and compared them to the corresponding response marginals computed for natural image patches. Secondly, we evaluated the quality of the full model by performing inference on natural image patches and judging the plausibility of the results. Details of the sampling and inference experiments can be found in sections 4.5.2.2 and 4.5.2.3 below.

4.5.2.2 Results: Plausibility of samples from the masked RBM

The full samples from the masked RBM are shown in Fig. 4.11, right. Though they do not exhibit as much structure as true natural image patches (Fig. 4.11, left), the presence of multiple sharp edges makes them look much more convincing than the typical blurred samples one may obtain from a single RBM. Moreover, the samples clearly capture important characteristics of the training patches (such as the dominance of homogeneous regions and the shape of the boundaries of these regions), despite the relative simplicity of the model and the fact that K was chosen to be small.

In order to further assess the quality of the samples from the masked RBM we compared the statistics of responses of different types of filters with the filter responses for natural image patches. Filter response marginals were computed for a bank of 24 even and odd Gabor filters and a set of 24 random zero mean filters (size 7×7 pixels (a random subset of the filters is shown in Fig. 4.12 as insets). Before computing the filter responses, we converted all the patches to gray scale. In order to provide a baseline we computed filter response marginals for natural image patches and samples from the masked RBM but also for samples from a single, unmasked Beta RBM (the same that was used as appearance model in the masked RBM) as well as from a Gaussian with covariance matched to the natural image patches.

The results (displayed as log-probability of each response value) for a random sub-

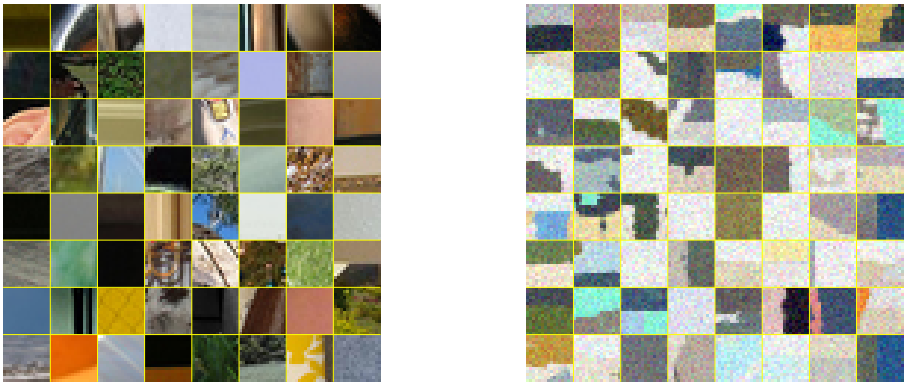


Figure 4.11: True natural patches (left) and samples from the masked RBM (right).

sets of the filters are shown in Fig. 4.12. For all filters the response histograms of samples from the masked RBM (in blue) have much heavier tails than those for patches sampled from the unmasked RBM (in red) or from the Gaussian model (cyan), but they are similar to the responses obtained from real image patches (green). There is one systematic mismatch between natural image patches and the samples obtained from the masked RBM: Due to the pixel-independent noise model (the visible units of the appearance RBM are conditionally independent given the hidden units) the peak of the histograms at 0 is underestimated for the samples from the masked RBM (this is because nearby pixels have an extremely low probability of having the same value, unlike true image patches). However, if we replace samples from the appearance model with the mean activations of the visibles given the binary hiddens in the last step of the Gibbs chains[6] and use those when composing the full, layered samples from the masked RBM we get the filter responses shown in Fig. 4.13 (only the region near the origin is shown). The tails remain largely the same but the peak at 0 is more pronounced, closely matching the ones obtained with true image patches. We would like to emphasize that the model has never been trained directly to match the statistics of natural images. Nevertheless, it reproduces some of their distinguishing features quite reliably. The improved matching, in particular the heavy tails, arose naturally with the use of a mask.

[6]That is we run a Gibbs chain in the top-layer of the appearance DBN for the same amount of time (5000 steps) sampling all units in each step. Only during the last step we use the mean activation of the visible units $\hat{v}$ given the binary hidden states when generating a shape using the conditional distribution defined by the beta RBM APP($\hat{\mathbf{v}}|\mathbf{h}^{(a)}$) (rather than a sample).

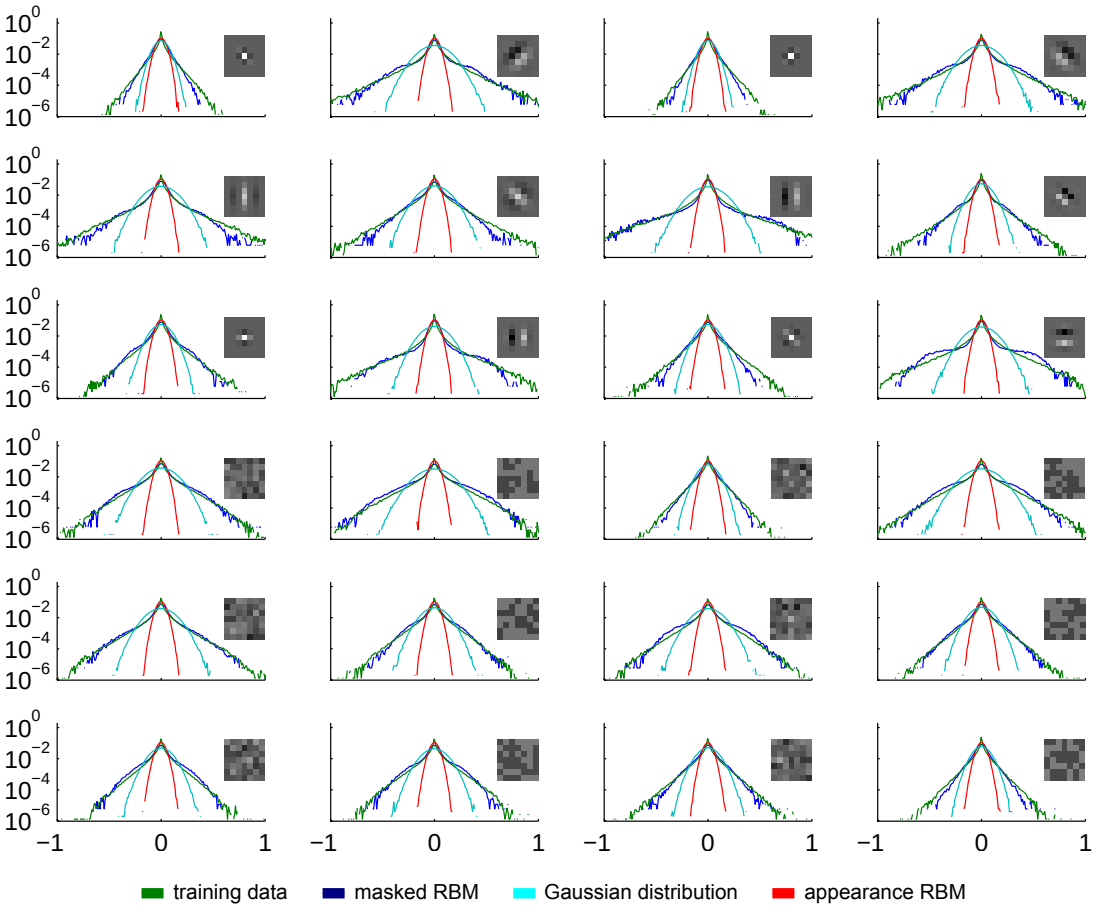


Figure 4.12: Filter responses for various kinds of patches: **Green**: real image patches. **Blue**: samples from the masked model with $K = 3$. **Red**: samples from the appearance model only (no shape). **Cyan**: Gaussian noise with the same covariance as the real patches. For each histogram the corresponding filter is shown as an inset. Whereas the samples generated from a single RBM exhibit a Gaussian-like response, the response obtained for samples from the masked RBM closely match those obtained from real image patches.

4.5.2.3 Results: Patch segmentation & depth inference

The goal of this experiment is to investigate whether learning an efficient representation of the data leads to the model being able to reason about image regions and relative depths. For this purpose we chose a simple scenario shown in Fig. 4.14: patches that contained simple shape-based depth cues were extracted from an image (a). For each patch, the model inferred a segmentation mask $\mathbf{m}$ with up to K=3 regions (b.1), a relative depth ordering π (front to back: red — green — blue), the potentially partially

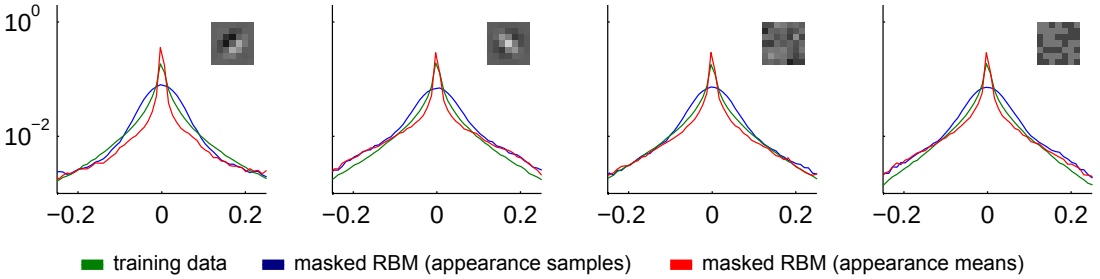


Figure 4.13: Difference between sampled and mean activations in a zoomed-in region close to the peak, for the first four Gabor filters. **Green**: real image patches. **Blue**: samples from the masked model with $K = 3$ where the activations of the visible units have been sampled given the binary hidden states. **Red**: samples from the masked model with $K = 3$ where the activations of the visible units are the average of the activations given the binary hidden states. Due to the smoothness induced by the averaging, the peak at 0 is much more pronounced and is much closer to the one obtained with real image patches. Similar results were obtained for the other Gabor filters.

unobserved shapes $\mathbf{s}_k$ of the two front-most layers (b.2) and the appearances $\hat{\mathbf{v}}_k$ of the three layers. For the examples shown, the model inferred segmentations, depth orderings and latent shapes largely consistent with the full image. Furthermore, the inferred latent shapes allow for removing the foreground shape and imputing the missing parts of the second layer shape (c.1 and c.2: segmentation mask with two layers and imputed image respectively).

Inferring relative depth using very local shape information only (such as provided by our 16×16 patches) is a highly ambiguous problem in the general case, not just for a computational model but also for human observers. Accordingly, the confidence of the model with respect to the relative depth of the regions in a patch can vary significantly between patches. For the examples shown in Fig. 4.14 the model is rather confident with respect to the inferred depth for patches 1, 2, 4, and 5, but considerably less confident for patch 3 (inference is performed by sampling from the posterior distribution; Fig. 4.14 shows the most likely depth ordering under the model for the five patches). More generally, the fact that the model is able to perform such a task at all might be surprising considering that it has only been trained on individual image patches without any built-in prior (e.g. about smooth boundary shapes) or additional information, such as the context (the larger shapes that the fragments in the patch are

part of), stereo data, or temporal information. Nevertheless, there are at least two plausible "cues" acquired by the model during training that are driving the results in Fig. 4.14. One relatively naïve cue the model uses is that it prefers to place smaller regions in the foreground. More importantly, however, it also prefers to explain image patches in terms of extended, roughly horizontal or vertical shapes. This behavior is rather robust and observed for all five examples in Fig. 4.14, in particular so for patch 3. It allows the model to complete the occluded shapes in a plausible manner and thus drives depth inference.

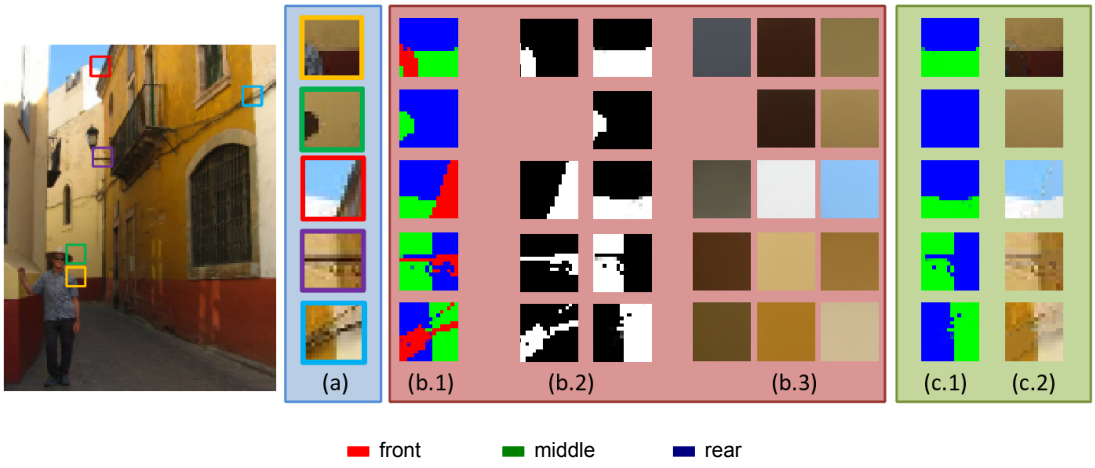
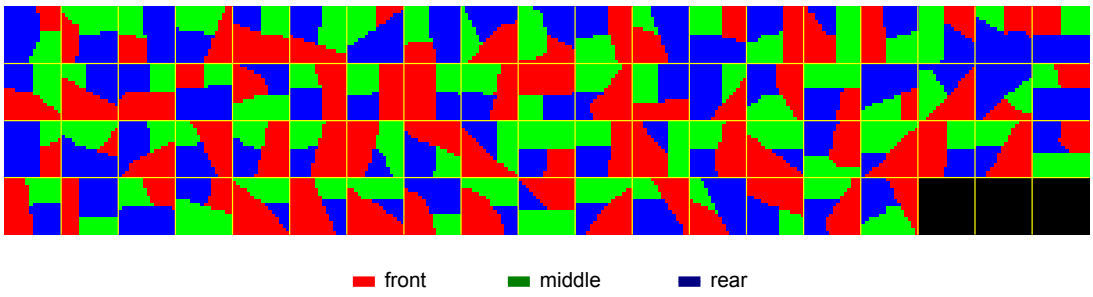


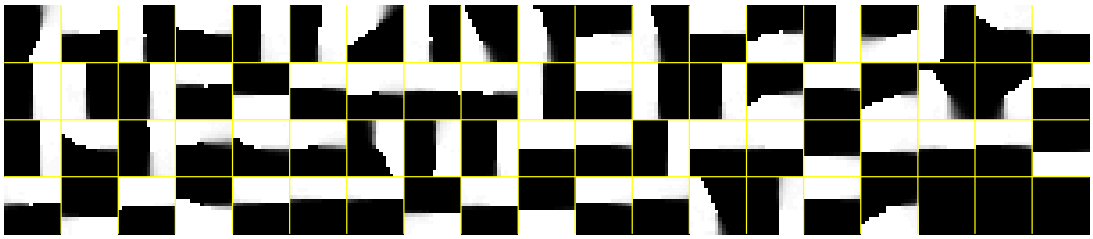
Figure 4.14: Inference for natural image patches: For a set of natural image patches (a), we ran full inference in the masked RBM with occlusion shape model. Inference produces a segmentation of the patch, i.e. the mask $\mathbf{m}$ (b.1), a depth ordering ($\boldsymbol{\pi}$) of the layers (color code in b.1: red - front, green - middle, blue - rear), the latent shapes ($\mathbf{s}_k$) of the 2 front-most objects (b.2) and the 3 latent appearances ($\widehat{\mathbf{v}}_k$; shown in b.3). This latent representation of the patch allows us to perform some simple image manipulation experiments: For instance, we can remove the foreground layer and knowledge of shape and appearance of the occluded layers allows us to complete the patch in a plausible manner. (c.1) shows the mask image after removal of the foreground; (c.2) shows the full patch with the area previously occupied by the foreground region filled in.

Further evaluation of depth inference: To evaluate the behavior of the model on a larger data set and to demonstrate how learning of a shape prior can drive depth inference we ran depth inference on 73 three-region *mask patches*, similar to patch 3 in Fig. 4.14, extracted from the segmentation images provided with the Berkeley

segmentation database.[7] Depth inference was run for 8000 iterations and the inferred depth after each iteration was recorded. For each patch we determined which of the three mask-regions was most frequently sampled to be the front-most region and which of the remaining two layers was most frequently chosen to be the middle layer. For the preferred middle-layer region we then determined, for each patch, the average shape fantasy associated with that region being the middle layer. The results are shown in Fig. 4.15a,b.



(a)



(b)

Figure 4.15: **Depth inference for mask patches** (a) Mask patches with three regions extracted from segmentation images from the Berkeley Segmentation Database. Each region is colored according to the depth inferred by the model as in Fig. 4.14: red - front; green - middle; blue - back. (b) Average shape fantasy for the middle (green) layer for the each of the mask patches and the associated preferred ordering shown in (a). Even though there is some variability, the model tends to explain the mask patches in terms of extended shapes overlapping each other, in many cases consistent with human judgment.

Although there is some variability, the model has a very clear tendency to explain the mask patches in terms of extended shapes overlapping each other, in particular in

[7] We used mask patches, i.e. patches for which the segmentation had already been provided, in order to separate depth inference from the segmentation problem. As explained in the main text the segmentation of a patch can be affected e.g. by matting or shading which is currently not handled well by the appearance model.

terms of roughly horizontal or vertical shapes. This is consistent with the results shown in Fig. 4.14 and a very plausible behavior given the training data in which regions of such shapes occur frequently (note that these shapes also feature prominently in the samples shown in Fig. 4.11). This behavior is also in rough agreement with the judgment of human observers: We showed the same 73 patches to 5 subjects and asked them to indicate, for each patch, which of the 3 regions they thought to be in front. The depth inferred by the model was consistent with the majority of human observers in 44/73 cases, i.e. for 60% of the patches which is a considerably higher percentage than expected if the model selected the front-most region randomly (a random choice of the front most region in this task would correspond to an agreement of 33%). At the same time, human subjects were in agreement with each other only for 32/73 patches (44%) highlighting the general difficulty and ambiguity of this task. Note that these results cannot be explained by a simple bias of the model to place smaller regions in the front since for 37/73 (51%) of the test patches the region that was inferred to be in front by the model was in fact the largest of the three regions while the smallest region was inferred to be in front only in 17/73 (23%) cases. Overall the model behavior seems reasonable given that such roughly horizontal and vertical shapes are particularly frequent in our training data so that representing e.g. patch 3 in terms of such shapes is a likely explanation in light of this training data. Thus, learning an efficient representation of the data also has made the model pick up certain simple depth cues, despite never having received any kind of depth information with the training data.

Further evaluation of image segmentation: Although inference for natural image patches typically leads to plausible segmentations, there are currently two main limitations of the model: Firstly, the model has difficulties correctly segmenting image patches that exhibit matting or shading, since this is not accounted for by the model (this effect can be observed, for instance, for patch 4 in Fig. 4.14). Also, the model currently does not have a suitable prior over the number of regions, so it has a tendency to over-segment patches which have fewer than K coherent regions (such as the second patch in Fig. 4.14). Figures 4.16a and 4.16b illustrate these effects: Figure 4.16a shows the segmentation inferred for patches that were chosen to be largely homogeneous, i.e. so that it should be possible to explain them using a single layer. Figure 4.16b shows the segmentation inferred for patches that were chosen to contain two regions. For each patch, several samples of the inferred masks are shown which were obtained by running inference several times using different random initializations of the mask. For both sets of patches the model has a tendency to over-segment the

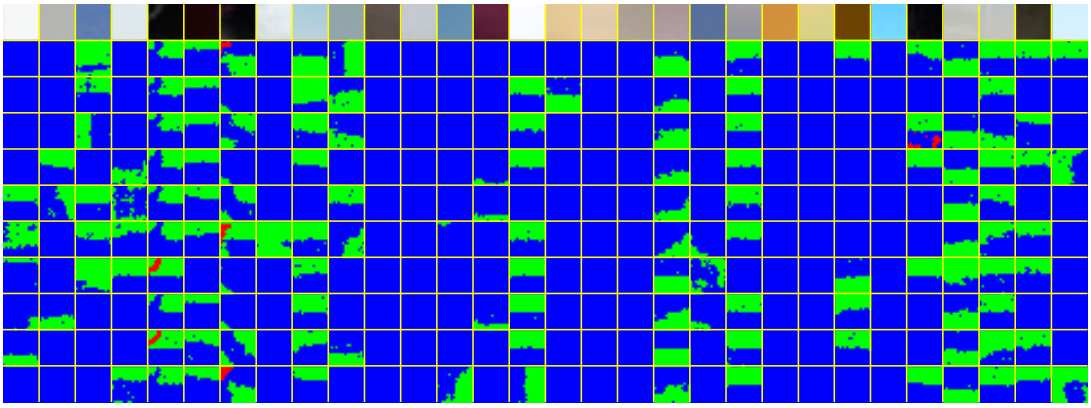
patches, effectively hallucinating region shapes into homogeneous areas of the patches. For the two region patches in Figure 4.16b the model has the additional tendency to "abuse" the third, unneeded region to explain pixels of intermediate colors which tend to occur at region boundaries due to matting (e.g. patches 2 and 9 in Fig. 4.16b), and also to over-segment regions which exhibit slight shading (e.g. patches 8 and 10 in Fig. 4.16b). This second phenomenon in Fig. 4.16b can partially be explained by the appearance model having acquired a strong preference for homogeneous colors (see also discussion in Le Roux et al. (2011)), however, the more general problem is the lack of an appropriate mechanism for selecting a suitable K for a given patch. Incorporating a prior over K effectively corresponds to model selection and is non-trivial since we cannot compute the normalization constant of either the appearance or shape RBM. This issue is discussed further in section 4.6 below.

4.6 Discussion

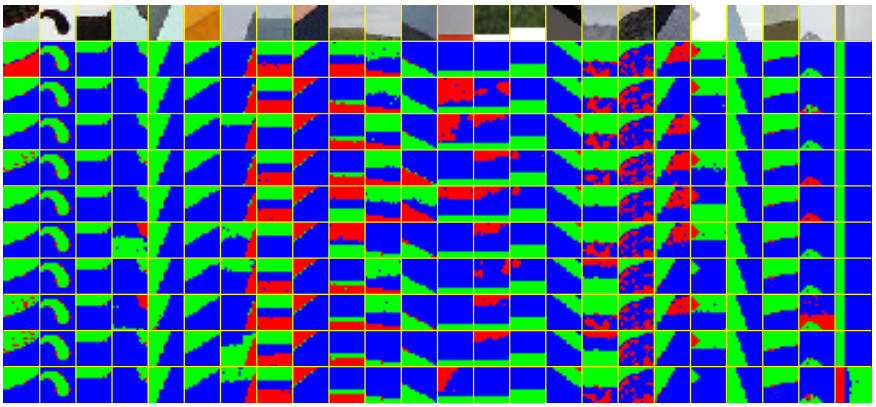
In this chapter we have proposed a model of region shape for the masked RBM in which shapes are drawn independently from a shape prior and are then combined according to a depth order in an occluding manner to partition of the image into regions. The shape prior is a binary RBM and is thus able to learn complex distributions of shapes. We have developed a Gibbs sampling scheme that allows for efficient inference and learning. We have further explained how this model of the mask can be combined with the masked RBM to give rise to a model in which an image patch is formed from several occluding objects, defined in terms of shape and appearance where shape and appearance are modeled independently, which provides an efficient way of dealing with the enormous variability of shapes and appearances in natural images.

We have evaluated the mask model on toy data (mask images) and have demonstrated that it can learn about the ground-truth shapes despite the presence of occlusions, and we have further found that the model is more efficient than the simpler "softmax"-model when the data is indeed generated from occluding shapes. We have trained the full masked RBM on natural image patches and have obtained a model that captures certain properties of natural images well (in particular the presence of largely homogeneous regions separated by relatively smooth boundaries), and, we have found that the model learns a plausible shape prior without any shape information being provided during learning.

The rich shape and appearance priors employed by our model, and its ability to



(a)



(b)

Figure 4.16: **Additional segmentation results**: The figure shows segmentation results for patches for which one would expect a segmentation into fewer than $K = 3$ regions. The top panel shows the results for 30 largely homogeneous (single-region) patches, the lower panel for 24 patches with two regions. For each patch the segmentation results for 10 independent restarts of inference with different initializations of the mask are shown. The model has a tendency to over-segment patches that for a human observer appear to contain fewer than $K = 3$ regions.

reason about occlusion for individual images makes the model sufficiently flexible so that it can be applied to general natural image patches. This distinguishes it from previous models that explicitly account for occlusion and which are largely limited to a small number of objects (see discussion in section 4.4 above). When applied to natural image patches the model is able to capture the presence of multiple regions separated by sharp boundaries which is an important feature of natural images. This feature has so far posed a challenge to other approaches used to model the structure in generic natural image patches except for the recent work work by Ranzato et al.,

2010a; Ranzato and Hinton, 2010 (see also discussion in section 4.4.1). One feature of natural image patches that the model currently does not capture well is the presence of textured regions (see also discussion below). Furthermore, it appears that the shape model has a slight bias towards horizontal and vertical shapes.

4.6.1 Limitations of the masked RBM

Nevertheless, the masked RBM with occlusion shape model has several shortcomings: Firstly, even though we have outlined a relatively efficient inference scheme, inference remains expensive, especially compared to standard RBMs. In particular the complexity of depth inference as described in section 4.3.3 is factorial in the number of layers K and represents one of the main bottlenecks of the model and currently effectively limits the model to $K \leqslant 4$. Although such K seems sufficient for small image patches such as the ones considered in the experiments in section 4.5.2, for larger images a larger K would most likely be needed. While this is currently not practical with the masked RBM, in chapter 5 we will describe an extension of the model that sidesteps this problem by representing a large image with a large number of small occluding objects.

A second shortcoming of the model is that evaluation is very difficult. The natural criterion for assessing model quality, the marginal probability of the data under the model (i.e. after integrating out the latent variables), is not applicable, because the intractability of the normalization constants of the RBMs involved, and also because of the presence of latent variables $(\widehat{\mathbf{v}}_{1\ldots K}, \mathbf{m}, \mathbf{s}_{1\ldots K})$ that are intractable to be integrated or summed out exactly. In the experiments above we have therefore used surrogate measures. Although this is a problem that is not unique to the masked RBM and applies to many undirected and latent variable models, it is certainly not a very satisfying situation and deserves further research (see Le Roux et al., 2011 for a discussion).

A third and related problem is the question of how to select K for a given image patch. As demonstrated in the experiments on natural image patches choosing K too large tends to result in an oversegmentation (cf. Fig. 4.16). This has two implications for learning: Firstly, during learning it allows the appearance model to acquire a strong preference for homogeneous regions since even e.g. simple gradients can be segmented into two (or more) regions with largely homogeneous colors. This is likely to be part of the reason why samples from the appearance model exhibit relatively little structure. Secondly, since the shape model uses the inferred segmentations and resulting latent

shapes as training data, oversegmenting a homogeneous region leads to the model effectively learning from its own hallucinations. This might be contributing to the slight bias towards horizontal and vertical shapes observed in the samples from the shape model trained on natural image patches. For these reasons it would be highly desirable to be able to select K for a given image patch not just at test time but in particular during learning. Selecting or inferring K is a difficult model selection problem. A principled approach would be based on the marginal probability of an image patch under models for different Ks (possibly taking some prior over the distribution of number of regions into account). Unfortunately, as explained above, the marginal likelihood cannot be computed because the normalization constants of the shape and appearance RBMs are unknown, and because it would require integrating out the latent variables.

4.6.2 Future Work

There are several directions for further research, some of which relate to limitations of the model discussed in the previous section. One of the major limitations of the masked RBM, the fact that it is currently restricted to small image patches, will be addressed in next chapter.

4.6.2.1 Alternative Inference Schemes

One interesting question would be to investigate alternative inference schemes beyond Gibbs sampling, in particular deterministic approaches. Within the existing Gibbs sampling framework it might be possible to develop a more solid scheme for depth inference based on a Metropolis-Hastings procedure that involves depth-moves similar to the split-merge MCMC technique for mixture models proposed by Jain and Neal (2007). In order to reduce the computational complexity of inference it might be possible to employ discriminative techniques, similar to, for instance, Salakhutdinov and Larochelle (2010) or Tu et al. (2001). Salakhutdinov and Larochelle employ a recognition network (see also Dayan et al., 1995) to obtain better initializations for mean field inference; Tu et al. use data driven proposals in a complex MCMC scheme. It is also conceivable that inference can be made faster by considering only a subset of possible depth orderings in each step (e.g. move a particular layer one position up or down in the depth order).

4.6.2.2 Inferring K

A problem that is important to address is the question of inferring K as this is also likely to improve the learned shape and appearance models. One major challenge encountered in this context is that inference of K, for a given patch, needs to be reasonably fast in order to be useful during learning. We have investigated various approaches, and one approach that appears to work relatively well in practice is related to the greedy learning approach developed by Williams and Titsias (2004): The masked RBM can be extended to include, for each pixel, an outlier component that accounts for image pixels not well explained by any of the K layers. In this formulation the image becomes a pixel-wise mixture of the outlier model, e.g. a pixel-wise independent uniform distribution and the masked RBM, so that, with a uniform mask model, the joint distribution over all variables involved (the equivalent to equation 4.1) becomes :

$$p(\mathbf{v},\mathbf{m},\mathbf{o},\hat{\mathbf{v}}_{1\ldots K},\mathbf{h}^{(a)}_{1\ldots K}) = p(\mathbf{o})p(\mathbf{m})\prod_{k=1}^{K} p(\hat{\mathbf{v}}_k,\mathbf{h}^{(a)}_k)\prod_i \left[U(v_i)\right]^{o_i}\left[\delta(\hat{v}_{m_i,i}=v_i)\right]^{1-o_i} \tag{4.39}$$

where $U(v_i)$ is the pixel-wise outlier distribution – e.g. a uniform distribution over $[0,1]$, and $o_i \in \{0,1\}$ an outlier indicator so that the pixel is explained by the outlier model if $o_i = 1$. $p(\mathbf{o}) = \prod_i p^{o_i}(1-p)^{1-o_i}$ is a pixel-independent prior specifying the prior probability p of the pixel being assigned to the outlier model.

This then allows for a greedy inference scheme in which inference for a patch is performed with $K = 1$ first, and only if too many image pixels are assigned to the outlier component is K being increased. This scheme allows selecting K and has the additional advantage that inference only ever needs to be performed with $K = 1$ for most patches (a large fraction of image patches is indeed homogeneous and thus well modeled by a single layer), thus speeding up inference across the dataset. An obvious question that arises in this approach is the question of how to select the prior probability of the outlier component, and the threshold for the number of pixels assigned to the outlier model that triggers the introduction of an additional layer (i.e. that leads to an increase in K). In preliminary experiments we have been investigating a scheme in which the threshold was set manually (10 pixels for 16×16 patches) and in which the outlier prior probability was determined by generating multi-region samples *with varying* K from the full masked RBM (sample generation is performed as described in section 4.5.2.1 above; i.e. we generated a set of image patches for which the ground-truth K was known), and by then performing a line search over different prior probabilities and choosing the one in which the correct K was inferred most frequently. Although

this scheme is to some extent ad-hoc, in preliminary experiments it appears to work relatively well. Also, encouragingly, the exact value of the prior probability is only moderately important. Training the masked RBM while inferring K for the training patches appears to lead to an appearance model that exhibits more structure than the appearance model used in the experiments above.

4.6.2.3 Models for foreground-background segmentation

In the formulation of the masked RBM as presented above all layers are equivalent in that they are governed by the same shape and appearance models. Furthermore, shape and appearance are modeled independently. While these assumptions are reasonable when modeling generic natural image patches, the model can relatively easily be modified to handle images with qualitatively different layers, and also to model shape and appearance jointly. Indeed, in Heess et al. (2011) we have shown how the model can be set up to represent a class of foreground objects in front of cluttered backgrounds.

In this version of the masked RBM we set $K = 2$, and the shape and appearance of the foreground objects are modeled jointly by one layer, while the background layer has a qualitatively different appearance model; the depth order of the layers is fixed. The joint model of shape and appearance of the foreground is a RBM with two sets of visible units, a set of binary units that model the binary shape image, and a set of continuous valued units for the appearance (Beta units as for the appearance model used in the experiments above) with energy:

$$E_{\text{mixed}}(\mathbf{v},\mathbf{m},\mathbf{h};\Theta) = E_{\text{Bin}}(\mathbf{m},\mathbf{h};\Theta^{\text{S}}) + E_{\text{Beta}}(\mathbf{v},\mathbf{h};\Theta^{\text{A}}) \tag{4.40}$$

where $E_{\text{Bin}}(\mathbf{m},\mathbf{h};\Theta) = \mathbf{m}^T W\mathbf{h} + \mathbf{b}^T\mathbf{m}$ is the energy function of a binary RBM (cf. chapter 2, section 2.2.4) and E_{Beta} is the energy of the Beta RBM as given by equation 4.38 (note that there are no biases for the hidden units in E_{Bin} since these are included in E_{Beta}).

The probability of an image $\mathbf{v}$ under this model is then given by:

$$P(\mathbf{v}) = \sum_{\mathbf{m}} \int \mathrm{d}\mathbf{v}^{\text{B}} \int \mathrm{d}\mathbf{v}^{\text{F}} \left(\prod_i \delta\left[v_i^{\text{F}} = v_i\right]^{m_i} \delta\left[v_i^{\text{B}} = v_i\right]^{(1-m_i)} \right) p_{\text{FG}}(\mathbf{v}^{\text{F}},\mathbf{m}) p_{\text{BG}}(\mathbf{v}^{\text{B}}) \tag{4.41}$$

where $\mathbf{v}^{\text{F}}$ and $\mathbf{v}^{\text{B}}$ are the latent appearances of the foreground and the background respectively, $p_{\text{BG}}(\mathbf{v}^{\text{B}})$ the marginal distribution of the background appearance RBM and $p_{\text{FG}}(\mathbf{v}^{\text{F}},\mathbf{m}) = 1/Z \sum_{\mathbf{h}} \exp\left\{-E(\mathbf{v}^{\text{F}},\mathbf{m},\mathbf{h})\right\}$ the marginal distribution over the visible

units of the joint foreground appearance and shape RBM. Note that this is effectively a two layer masked RBM with a particular kind of shape model.

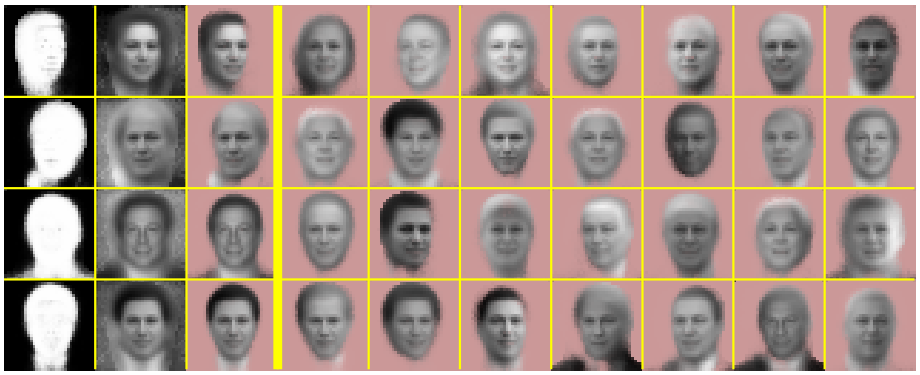
We have applied the model to a modified version of the "Labeled faces in the wild-A" (LFW-A)-dataset (Huang et al., 2007; Wolf et al., 2010) and demonstrate how the joint shape and appearance of faces can be learned directly from cluttered images with only weak supervision: We pre-train the background model on a set of general natural images patches, and subsequently train the foreground RBM (shape and appearance) in the context of the full model (equation 4.41) on images containing the foreground objects (see Fig. 4.17a for examples). We do not provide any additional information (such as ground-truth segmentations) with these training images, but the pre-trained background model is sufficient to learn the foreground model without further supervision by learning about the consistencies across the training images of regions that are *not* well explained by the background model (and thus foreground candidates). After training, the learned foreground model is able to generate joint samples of matching shape and appearance of faces (see Fig. 4.17b), and the full model can be applied e.g. to foreground-background segmentation tasks (Fig. 4.17c). We find that representing the foreground object independently of the background can be beneficial in recognition tasks. See Heess et al. (2011) for additional experiments and details.

One interesting extension of this model would be to include a third layer, placed in front of the foreground which would be able to account for the situation when the foreground object is occluded.

4.6.2.4 Richer training data

In the experiments described in section 4.5.2 we have used an unsupervised scheme to learn about shape and appearance from static natural image patches. In this context, but in particular also in scenarios such as the foreground-background model outlined in the preceding section (4.6.2.3) it would be very interesting to investigate the use of alternative datasets that provide additional information e.g. with respect to the segmentation. Such data sets could include human annotated data (e.g. data sets that include some pre-segmented images; the models proposed in this chapter are naturally suited for semi-supervised learning), data with associated depth information e.g. from the depth sensors, and especially spatio-temporal data that allows, for instance, to compute optical flow which could then be used as an additional cue when inferring the segmentation of an image (patch).

(a)



(b)

(c)

Figure 4.17: Example results for the foreground-background model from Heess et al. (2011). *(a)* Examples of the training data. *(b)*: Samples from the learned model. For the first three columns the format is similar to Fig. 4.8b, they demonstrate how shape ($\mathbf{m}$, left) and appearance ($\mathbf{v}^{\mathrm{F}}$, middle) combine to the joint sample (right). The remaining columns show further samples from the model. For the joint samples the red area is not part of the object. *(c)*: Inferred masks $\mathbf{m}$ (foreground-background segmentations) for a subset of the test images. Mask are superimposed on the test images in red. In most cases the model has largely correctly identified the pixels belonging to the face. Test images for which the model tends to make mistakes typically show the head in extreme poses. Labeling of the neck and the shoulders is somewhat inconsistent, which is expected given that there is considerable variability in the training images and that the model has not been trained to either include or exclude such areas. The same applies if parts of a face are occluded, e.g. by a hat.

Chapter 5

Modeling images with regions: The field of masked RBMs

Chapter 4 describes the masked RBM with occlusion-based shape model. The model is evaluated on various datasets consisting of small image patches. In this chapter we will discuss how to extend the work presented in chapter 4 to efficiently model large images.

One naïve way to extend the masked RBM to larger images would be to simply train larger appearance and shape models and to use a larger number of layers (K) (as there are likely to be more independent objects in a larger image than in a small patch). This is, however, problematic for several reasons. Firstly, larger appearance and shape RBMs would be required, which strongly increases the number of parameters and it is computationally expensive. Secondly, and more importantly, depth inference becomes quickly intractable as the number of layers grows (recall that the number of depth orderings that need to be explored is factorial in the number of layers K). In addition to these purely computational arguments the fact that images exhibit at least some degree of stationarity suggests that it would be desirable to incorporate a form of translation invariance into the model. Finally, the probability of image pixels belonging to different objects (or different parts of objects) increases with their spatial distance so that allowing for arbitrarily large shape and appearance models might not be the best use of computational resources.

In this chapter we therefore discuss an alternative way of extending the masked RBM to larger images: the field of masked RBMs which arises from replicating the masked RBM for small image patches at many positions across a larger image. When the occlusion-based shape model is integrated into the field of masked RBMs one ob-

tains a model that describes images in terms of many small, partially overlapping (and occluding) "objects" which are independent of each other, and each of which is governed by a shape and an appearance model, bearing an interesting resemblance e.g. to the dead-leaves model of Lee et al. (2001) (see also Jeulin, 1997). The resulting model is not only computationally considerably more efficient than simply scaling up the masked RBM, it also gives rise to a form of coarse translation invariance, incorporates an independence assumption regarding distant regions, and has an appealing interpretation in terms of a "superpixel algorithm".

The rest of this chapter is structured as follows: In section 5.1.1 we will describe the basic field of masked RBMs. The original idea for this basic form of the field of masked RBMs (FoMRBM) has been developed by Nicolas Le Roux and John Winn. Sections 5.1.2 and 5.1.3 discuss how the occlusion model can be integrated with the basic FoMRBM model, giving rise to a full generative model of mid-level natural image structure, and also describe how this model can be trained. Section 5.2 discusses related work. In section 5.3 we evaluate the model on two datasets: Section 5.3.1 describes experiments on a toy data set that demonstrate the general viability of the occlusion shape model in the context of the masked RBM. Section 5.3.2 applies the model to natural images, demonstrating that the model can learn about the rich structure present in natural images. We will discuss the limitations of the FoMRBM and its possible extensions in section 5.4. In particular, we will discuss a recursive hierarchical formulation of the model in this section.

Contribution

The original idea for the FoMRBM has been developed by Le Roux & Winn. My contribution is the occlusion shape model for the FoMRBM (the formulation of the model, inference, and learning) which made possible the experiments shown in this section, which are also my own work. The Deep Segmentation Network described in section 5.4.2 (Future Work) again is an idea originally developed by Le Roux & Winn. I have, however, conducted various experiments on this model. My work gives rise to the (very preliminary) illustrative results shown in section 5.4.2.

5.1 Model

5.1.1 Field of masked RBMs

The masked RBM described in chapter 4 models an image patch as a pixel-wise mixture of K fully aligned latent patches of the same size as the observed image patch. For each pixel in the observed image patch one of the K corresponding pixels in the K latent patches is chosen. For moderately large image patches and a small number of latent patches (i.e. for small K) inference and learning in the MRBM remains practical. In the experiments of section 4.5.2, for instance, the size of the patches that we applied the masked RBM to was 16×16 pixels and we chose $K = 3$. As discussed above, one way to model larger image patches would be to simply increase the size of the observed image and latent patches, and use a larger number of latent patches. Yet, as also discussed, this approach has several disadvantages, in particular it is computationally expensive (depth inference is factorial in K) and leads to a large number of parameters for the shape and appearance RBMs.

The FoMRBM takes a different approach in order to model large images: We keep the size of the latent patches small but use a much larger number of them. These small latent patches laid out across the image such that each pixel of the observed image is covered by several of the latent patches. That way, as for the masked RBM, each image pixel can still be explained by one out of several different latent patches, although, in this formulation, each latent patch covers only a small part of the observed image (e.g. a 16×16 pixel region), and different parts of the observed image will be covered by different latent patches. Using small latent patches to achieve a dense coverage of a larger image has several advantages and addresses most of the points raised above: It has computational advantages in terms of the size of the individual RBMs and especially in terms of depth inference (see detailed discussion below). Furthermore, it allows us to share parameters across latent patches at different positions in the image which reduces the number of model parameters that need to be learned. Also, it gives rise to a form of translation invariance since the same basic model structure will be replicated at many different positions across the image.

One obvious question that arises when adopting this scheme is how to lay out the latent patches to cover the image. One relatively simple scheme is to simply tile the image into non-overlapping regions of the size of the latent patches (say 16×16 pixels) and to model each of these regions using a masked RBM with K layers. This scheme, which is illustrated in Figure 5.1a, is conceptually very simple and it allows

for relatively efficient inference (we simply perform inference in each of the masked RBMs separately). A major downside of this scheme is, however, that it will lead to a poor representation if the structure in the image is not well aligned with the chosen tiling. The translation invariance achieved by this scheme is rather coarse: 16 pixels (or multiples thereof) horizontally or vertically.

A more flexible scheme that still allows for a computationally relatively efficient implementation of inference is obtained by realizing that there is no reason why latent patches would have to be aligned with each other. The pixel-wise mixture formulation allows us to lay out latent patches in a partially overlapping manner: In principle, any layout of the individual latent patches is possible, as long as each pixel of the observed image is covered by at least one latent patch. In practice it still seems reasonable to choose a layout in which each image pixel is covered by the same number of K latent patches (to allow the same degree of flexibility everywhere in the image). One way to achieve this is by laying out patches on K offset grids. Patches within the same grid are abutting and non-overlapping; but different grids are offset relative to each other so that patches in different grids are partially overlapping. For 16×16 patches and $K = 2$, for instance, there would be two grids that would be offset relative to each other by 8 pixels in the horizontal and vertical direction (this is illustrated in Fig. 5.1b); for $K = 4$ there would be four grids and these would be aligned with the image as follows: The first grid would be offset relative to the image boundaries by some amount horizontally and vertically. All the other grids would then be offset from the first grid horizontally and/or vertically by 8 pixels. For instance, choosing, for the first grid, an offset relative to the image boundary of 4 pixels horizontally and vertically, the other grids will then be aligned with image pixels $(4,12)$, $(12,4)$, $(12,12)$ (see Fig. 5.1c and also Fig. 5.2). In both scenarios (i.e. for $K = 2,4$) no two latent patches are fully aligned. The major advantage of this scheme compared to a simple tiling of the image into a single grid of non-overlapping masked RBMs is that it gives rise to a finer degree of translation invariance (e.g. 8 pixels horizontally and vertically, or multiples thereof, in the case of $K = 4$) than would be achieved otherwise. It should be noted that in this scheme all latent patches are independent and equivalent, i.e. they are all modeled by the same appearance model (appearance RBM) as was the case for the masked RBM described in the previous chapters. In particular, patches belonging to the same grid are not qualitatively different from patches belonging to other grids. The arrangement into grids simply ensures that each pixel is covered by the same number of latent patches. Also, we exploit the regularity of this layout in the formulation of the hierarchical

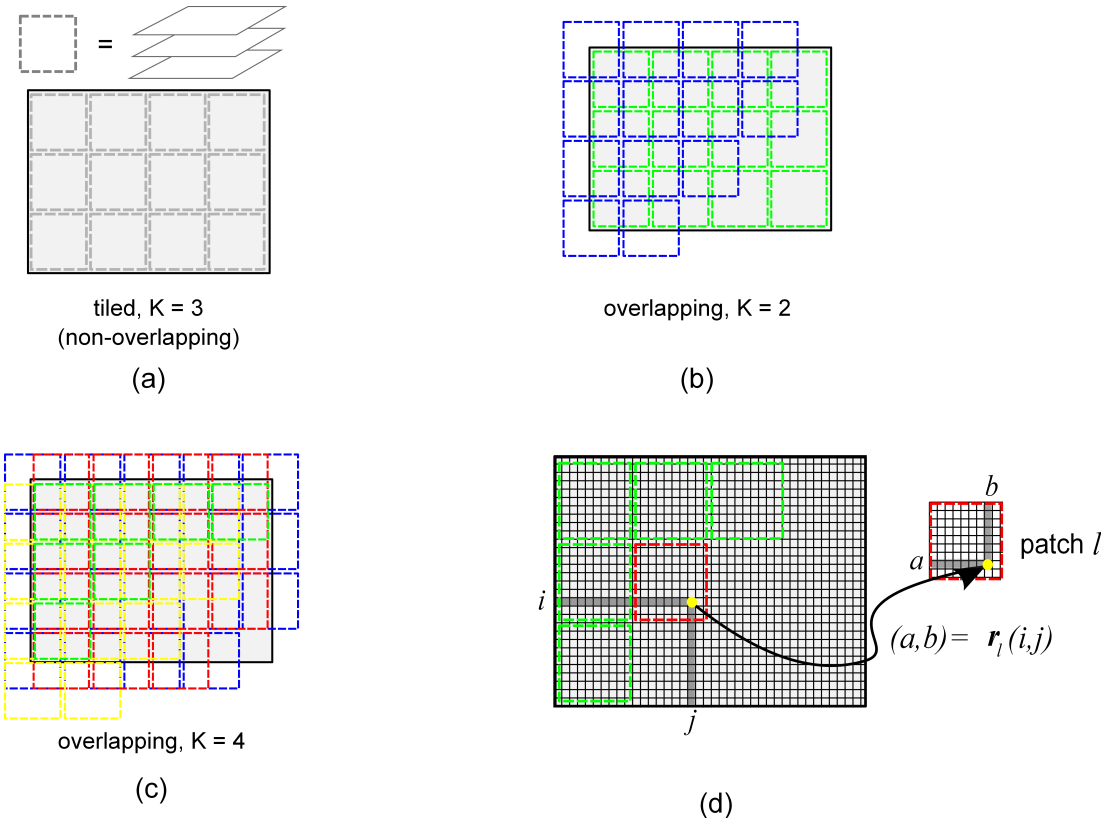


Figure 5.1: **Patch layout in the Field of masked RBMs** *(a)* Tiled layout. The image (shown in light gray in the background) is tiled into *non-overlapping* patches (dashed squares) of the same size as the latent patches. Each of these image patches is then modeled by a masked RBM with K latent patches that are aligned with the image patch (here, $K = 3$). *(b,c)* The image is covered by partially overlapping latent patches. Patches are laid out such that each image pixel is covered by K latent patches ($K = 2$ in *(b)*; $K = 4$ in *(c)*). Each patch is modeled by a shape and an appearance RBM as shown in Fig. 5.2. Patches are shown in different colors for visual clarity, but all patches are equivalent (i.e. the shape and appearance RBM are the same for all patches). *(d)* Illustration of the mapping of the index of an image pixel (yellow dot) onto the index of the pixel of one of the overlapping latent patches (shown in red; for clarity only one overlapping patch is shown). The function $r_l(\cdot)$ maps the index of the pixel with respect to the image boundaries (i, j) onto the corresponding index (a, b) of the pixel relative to the patch boundaries: $(a, b) = r_l(i, j)$, where l is the index of the latent patch shown in red.

extension of the model which we will explain in section 5.4.2

The joint distribution over the image and latent appearances conditioned on the

mask is given by

$$p(\mathbf{v}, \hat{\mathbf{v}}_{1\ldots K}, \mathbf{h}^{(a)}_{1\ldots K} | \mathbf{m}) = \prod_{l=1}^{L} \mathrm{APP}(\hat{\mathbf{v}}_l, \mathbf{h}^{(a)}_l) \prod_i \delta(\hat{v}_{m_i, r_{m_i}(i)} = v_i). \tag{5.1}$$

This is the equivalent of equation (4.1) for the masked RBM (cf. section 4.2 in chapter 4), where, as before, $\mathbf{v}$ is the image, $\hat{\mathbf{v}}_{1\ldots L}$ are the latent appearances, $\mathbf{m}$ is the mask, and the index i runs over all image pixels. There are, however, several differences: Firstly, $\mathbf{v}$, and $\mathbf{m}$ are much larger than any $\hat{\mathbf{v}}_l$. Secondly, there are $L \gg K$ latent patches, many more than there are latent patches overlapping with any particular pixel. Thirdly, $l = 1 \ldots L$ is the index over all patches. The mask is now an L-valued image $m_i \in 1 \ldots L$ but at each particular pixel it can only take one out of K values (corresponding to the indices of the patches that overlap with image pixel i). Finally, we need an explicit mapping between the index of an image pixel and the index of the corresponding pixel in a particular patch l. This is given by $r_l(i)$ which returns the index of the pixel of patch l that corresponds to image pixel i if l overlaps with i (otherwise $r_l(\cdot)$ is not defined). See Figure 5.1d for an illustration. In the above notation $\hat{v}_{l,j}$ refers to pixel j of patch l so that $\hat{v}_{m_i, r_{m_i}(i)}$ refers to that pixel of patch m_i which is aligned with image pixel i (m_i contains the index of the patch that is visible at image pixel i according to the mask; $r_{m_i}(i)$ provides the mapping from the index of the image pixel i to the index of the corresponding pixel in the patch m_i).

Inference in the FoMRBM is performed in exactly the same way as for the simple masked RBM (cf. Appendix B.1), with the only additional difficulty that one now needs to keep track, for each image pixel, which latent patches compete to explain this pixel.

5.1.2 Modeling shape and occlusion in field of masked RBMs

All mask models (i.e. the naïve uniform model, the softmax model, and the occlusion model) described in the previous chapter for the masked RBM can be used at the image level. As pointed out already for the masked RBM, the uniform model does not lead to a meaningful generative model of images, since it simply assigns a random mask value to each pixel independently. It can be used for inference (i.e. when inferring the mask for a given image) since in this case the mask will be largely determined by the image. Figure 18 in Le Roux et al. (2011) demonstrates, however, that even during inference using one of the two advanced models (softmax or occlusion) instead of the naïve uniform model yields better results: In particular these models usually lead to more coherent masks without significant loss in reconstruction accuracy.

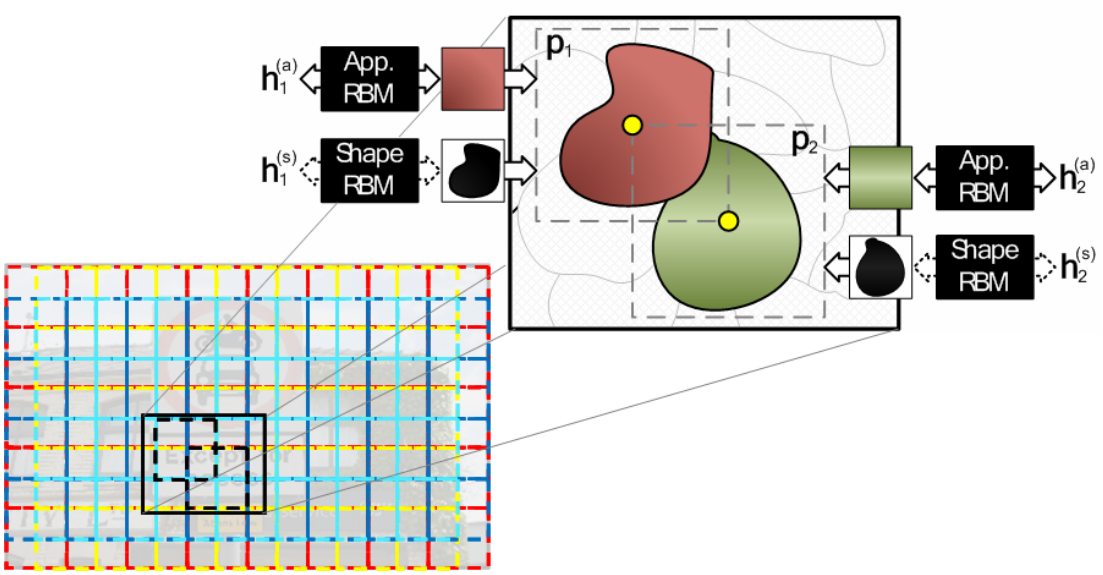


Figure 5.2: **A field of masked RBMs, where an image is represented using a set of overlapping patch models.** *Left*: the image is covered by K (here $K = 4$) grids of non-overlapping, abutting patches (each grid is shown in a different color: red, yellow, cyan, blue). The different grids are spatially offset so that the patch boundaries in different grids do not align and each pixel is covered by K partially overlapping patches that compete to explain the pixel. Note that all latent patches are independent and equivalent. The choice of different colors for patches in different grids only serves to keep the visualization uncluttered. *Right*: blow-up of the interaction between two overlapping patch models. Competition between patch models leads to a segmentation of the image into "superpixels", with one superpixel per patch. The appearance and the shape of each superpixel are modeled by separate RBMs. Figure courtesy of Nicolas Le Roux, John Winn, & Jamie Shotton.

The occlusion model leads to a particularly appealing interpretation at the image level: each patch model can be thought of as an independent expert modeling shape and appearance of an image patch. It consists of an appearance RBM that determines the color – or more generally texture – of a patch and a binary RBM that determines its shape, as is illustrated in the blow-up of Figure 5.2. An image is generated by covering it fully with such patches in an occluding manner. This generative process bears some resemblance to the "dead-leaves model" (Jeulin, 1997; Lee et al., 2001) although there are important differences (see section 5.2). The advantage of the occlusion model over the softmax model in this context is that under the occlusion model the shapes associated with the individual latent patches are independent whereas for the softmax

model there is no real notion of a patch-specific shape and binary RBMs of the different latent patches interact directly to form the mask.

The joint distribution over the mask, shapes, and depth ordering of the occlusion shape model for the FoMRBM is given as follows:

$$P(\mathbf{m}, \mathbf{s}_{1..L}, \mathbf{h}^{(s)}_{1..L}, \pi) \propto \quad P(\pi) \quad \left(\prod_i \delta(s_{m_i, r_{m_i}(i)} = 1) \prod_{l \in o(i): \pi(l) < \pi(m_i)} \delta(s_{l, r_l(i)} = 0) \right) \times \quad \left(\prod_l \text{SHAPE}(\mathbf{s}_l, \mathbf{h}^{(s)}_l) \right), \tag{5.2}$$

subject to the constraint that for each mask pixel only certain values out of $1 \dots L$ are valid (for each pixel the mask can only take the indices of those latent patches that overlap with that pixel). Note that this is effectively equation (4.12) with the main difference being that latent patches are now smaller and no longer aligned with the image so that we need to map image pixels onto patch-pixels and vice versa. As in (4.12) $\mathbf{s}_l$ is the l-th binary shape, $\mathbf{h}^{(s)}_l$ are the hidden units of the l-th patch, and π is the depth order. $o(i)$ is the set of all patches $l_1 \dots l_K$ that overlap with image pixel i (there will be K patches overlapping each image pixel if the latent patches are laid out as described in the previous section and illustrated in Fig. 5.1b,c and in Fig. 5.2). Note that as explained for equation (5.1), $s_{m_i, r_{m_i}(i)}$ refers to the pixel of shape patch m_i that is aligned with image pixel i: m_i selects the patch, and $r_{m_i}(i)$ maps the index of image pixel i onto the index of the corresponding pixel in patch m_i.

The generative process is also very similar to the occlusion model for the simple masked RBM:

- sample a depth ordering π
- for each latent patch $l = 1 \dots L$ generate an appearance $\hat{\mathbf{v}}_l$ by drawing L independent samples from the appearance RBM
- for each latent patch $l = 1 \dots L$ sample a shape $\mathbf{s}_l$ by drawing L independent samples from the shape RBM
- generate the mask $\mathbf{m}$ using the sampled shapes $\mathbf{s}_{1 \dots L}$ and the depth ordering π
- compose the image $\mathbf{v}$ from the mask $\mathbf{m}$ and the appearance samples $\hat{\mathbf{v}}_{1 \dots L}$.

Two points should be noted about this formulation: Firstly, the proportionality sign in equation (5.2) indicates an issue that we have discussed already in the context of

the masked RBM (see section 4.3.2 in the previous chapter): In the formulation of equation (5.2) the shapes are not truly marginally independent. This is because each image pixel has to be explained by one of the overlapping latent patches, so choices of shapes with all shapes off for one (or more) image pixels are invalid (i.e. situations in which $s_{l,r_l(i)} = 0 \ \ \forall l \in o(i)$ for at least one pixel i). Exact sampling from the model in equation (5.2) therefore requires sampling all shapes jointly.

For the masked RBM we achieved marginal independence of the shapes by assuming that the rear-most shape is always on, and this assumption could be taken properly into account during inference and learning (cf. discussion in section 4.3.2 of chapter 4). For the FoMRBM we will make a similar assumption, although the situation is slightly more complicated than in the masked RBM and requires some approximations: When generating from the FoMRBM we ensure that each image pixel is covered by at least one patch by forcing the shape of the rear-most patch to be on if that pixel is not covered by the shape of any other patch. In the masked RBM we assumed that the rear-most shape is drawn from a special shape model, and did not take the shape of the rear-most patch into account during depth inference and learning. The model therefore remained well defined and learning and inference remained exact. This is no longer the case for the FoMRBM: Since patches are partially overlapping, a single patch might be the rear-most patch for some image pixels that it is overlapping with but not for others so that we cannot modify the model in the same way as for the masked RBM. Instead, during depth inference and learning we currently simply ignore the fact that some shape pixels have been "forced". These steps are therefore only approximate in the way they are presented below. We will discuss a principled solution to this problem in section 5.4.1.1 below.[1]

The second point to notice about the above model relates to the nature of the depth ordering π. For the version of the FoMRBM with occlusion shape model presented here we assume a global ordering[2], although the relative depth-ordering of non-overlapping patches matters only insofar as the depth ordering induced by shared neighbors has to be consistent. This is illustrated in Figure 5.3. Configurations such as shown in Fig. 5.3a are valid, but the example in Fig. 5.3b is not. It should be noted, however, that while this is how we have chosen to implement the model that we have

[1] It is also possible to write down a well-defined generative model which makes the fact that the rear-most pixel can be forced to be on explicit. This is discussed in section C.1 in the appendix.

[2] In practice, we assign each patch a continuous depth value, this ensures that we can always find a new depth value that positions a particular patch in between two other patches. The actual value is, however, of no importance; only the induced ordering is relevant.

used for the experiments presented below, other variants are conceivable. For instance, it would be possible – and this might even be advantageous – to enforce only local consistency.

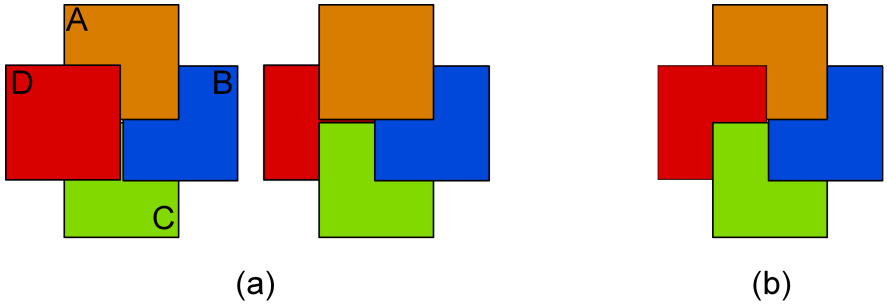


Figure 5.3: **Consistency of induced depth orderings**: *(a)* In both examples, patch B induces a relative ordering between non-overlapping patches A and C. The two relative depth orderings in *(a)* are valid in our current implementation of the model. *(b)* Example of an invalid depth ordering: The local depth orderings are not consistent since we have $A > B$, $B > C$ $C > D$ but $A < D$

5.1.3 Inference

The occlusion model still allows for efficient inference. It is performed by updating the mask, the latent shapes, the shape hidden units, as well as the depth ordering in an alternating manner as for the masked RBM.

Depth inference: In the case of the masked RBM there were $K!$ depth orderings that we needed to explore. Clearly, exploring $L!$ depth orderings of the $L \gg K$ latent patches in the FoMRBM would be computationally intractable (there are, for instance, 1218 patches in the experiment described in section 5.3.2). We are saved, however, by the fact that even though each image is explained by a typically large number of patches, each individual patch overlaps only with a small number of neighbors. Thus, instead of determining a global depth order of all patches (which would clearly be infeasible) it is sufficient to infer the depth of each patch relative to its neighbors. The depth of a particular patch given a fixed relative order of its neighbors can be determined following the principles described for the simple masked RBM in section 4.3.3 and the full local ordering of all patches covering the image is determined in an iterative manner by considering each patch in turn and updating its depth relative to its (overlapping) neighbors, keeping the depth ordering of the neighbors fixed:

Consider, for instance, the case with $K = 4$ and the global patch layout shown in Fig. 5.1c and Fig. 5.2, in which each image pixel is covered by four competing patch models. In this case each latent patch overlaps partially with 8 neighbors. Given a mask and an existing depth-ordering π we perform depth inference as follows: The latent patches covering the image are considered in a random order. For each latent patch we keep the relative ordering of its neighbors (and of all other patches) fixed and only re-infer the depth of the selected patch with respect to its neighbors. In the case at hand where each patch has 8 overlapping neighbors, nine relative depths need to be considered (including the possibility that the patch is in front or behind all its neighbors). For each of these relative depths we obtain a set of observed and a set of unobserved pixels, for the selected patch and for its neighbors as illustrated in Fig. 5.4. Unobserved shape pixels are filled in by masked Gibbs sampling as outlined in section 4.3.3 for the simple masked RBM. The (unnormalized) probability of a particular position is then obtained from the unnormalized log-probabilities of the completed shapes in the same way as described in section 4.3.3. Assume that the currently selected patch is l_0. Assume further that its N neighbors are patches $l_1 \dots l_N$. The unnormalized probability of a new depth ordering π' in which only the relative depth position of patch l_0 has changed is given as follows:

$$P(\pi'|\mathbf{m}, \hat{\mathbf{s}}^{\pi'}_{1..K}, \pi) \propto \prod_{l:l\neq l_0} \delta(\pi'(l) = \pi(l)) \prod_{n=0}^{N} \text{SHAPE}(\hat{\mathbf{s}}^{\pi'}_{l_n}) \tag{5.3}$$

where, as before, $\hat{\mathbf{s}}^{\pi'}_{l_n}$ is the latent shape corresponding to patch l_n with the unobserved pixels for depth order π' sampled from the posterior.[3] The superscript $^{\pi'}$ indicates the dependence of the completed shape on the depth ordering. Note the product of delta functions ensures that only the element l_0 of π is being updated. Algorithm 1 provides pseudo-code for re-sampling the relative depth of a patch l_0 with respect to its neighbors. Note that in algorithm 1 we use SHAPE($\mathbf{s}$) to refer to the unnormalized probability of a binary shape $\mathbf{s}$.

Two additional considerations make depth inference noticeably more efficient: Firstly, the main computational expense during depth inference arises from having to complete the partially observed shapes for different depth orderings of the patch of interest. Equation (5.3) suggests that we need to obtain completed shapes $\hat{\mathbf{s}}^{\pi'}_{l}$ for the selected

[3] As explained in section 4.3.3 the correct thing to do would be to marginalize over the unobserved variables. Since this is not possible we replace the sum with a sample from the posterior. This will lead to a biased estimate of the marginal likelihood but we have nevertheless found the approach to work well in practice.

Algorithm 1 Update relative depth for patch l_0

Require: input $\mathbf{m}$, l_0, $l_{1\ldots l_N}$, π

$\mathbf{o} \leftarrow$ indices of overlapping patches $l_1 \ldots l_N$ ordered according to depth π

{o_1 then contains the index of the rear-most patch of the neighbors, o_N that of the front-most}

{determine unnormalized log probability for all $N+1$ possible relative depths}

{patch l_0 behind all neighbors}

$\pi_0 \leftarrow \pi$, set $\pi_0(l_0)$ s.t. $\pi_0(l_0) < \pi(o_1)$

determine completed shapes $\hat{\mathbf{s}}_{l_0}, \hat{\mathbf{s}}_{l_1 \ldots l_N}$ given $\mathbf{m}$ and π_0

$p_0 \leftarrow \sum_{n=0}^{N} \log \mathrm{SHAPE}(\hat{\mathbf{s}}_{l_n})$ {compute unnormalized log probability}

{all intermediate positions for patch l_0}

for $i = 1$ to $N-1$ **do**

$\pi_i \leftarrow \pi$, set $\pi_i(l_0)$ s.t. $\pi(o_i) < \pi_i(l_0) < \pi(o_{i+1})$

determine completed shapes $\hat{\mathbf{s}}_{l_0}, \hat{\mathbf{s}}_{l_1 \ldots l_N}$ given $\mathbf{m}$ and π_i

$p_i \leftarrow \sum_{n=0}^{N} \log \mathrm{SHAPE}(\hat{\mathbf{s}}_{l_n})$ {compute unnormalized log probability}

end for

{patch l_0 in front of all neighbors}

$\pi_N \leftarrow \pi$, set $\pi_N(l_0)$ s.t. $\pi_N(l_0) > \pi(o_N)$

determine completed shapes $\hat{\mathbf{s}}_{l_0}, \hat{\mathbf{s}}_{l_1 \ldots l_N}$ given $\mathbf{m}$ and π_N

$p_N \leftarrow \sum_{n=0}^{N} \log \mathrm{SHAPE}(\hat{\mathbf{s}}_{l_n})$ {compute unnormalized log probability}

{determine new relative depth position}

sample new relative depth d according to $p(d) = \frac{\exp(p_d)}{\sum_{d'} \exp(p_{d'})}$

return π_d {return new depth order}

patch and for its N neighbors for each of the $(N+1)$ relative depth orderings. A naïve approach would thus require us to obtain $(N+1)^2$ completed shapes when re-inferring the depth of a single patch, i.e. $L(N+1)^2$ for a full sweep re-inferring the depth of all patches. Yet, the set of unobserved pixels in the neighboring patches depends only on whether the selected patch is in front or behind the respective neighbor. This considerably reduces the number of completed latent shapes that need to be considered for the neighbors: there are only two completed latent shapes for each neighbor (one for the case when the selected patch is in front of the neighboring patch, and one for the case when the selected shape is behind). Together with the $N+1$ completed shapes that need to be considered for the selected patch (one for each position of the patch relative to its neighbors) there are $3N+1$ in total. A full sweep through the set of all latent patches (L) thus requires the sampling of $L(3N+1)$ completed latent shapes.

Secondly, it should be noted that although it is conceptually convenient to think of updating the relative depth of a single patch at a time, in practice it is also possible to consider multiple latent patches simultaneously as long as the patches for which depth inference is performed in parallel do not have common neighbors (during depth inference the latent shape of the selected patch *and* of its neighbors are updated). For instance, for the layout for $K=4$ described above we can divide the set of all latent patches into 16 subsets, each of which contains only independent patches so that they can be updated simultaneously.

Updating the mask: Given an updated depth ordering π' and the corresponding set of the updated latent shapes, $\mathbf{s}'_{1\ldots L}$ the mask can then be updated in the same way as for the simple masked RBM[4]. First, a sample of the state of the shape hidden units is obtained for each patch

$$\mathbf{h}_l'^{(s)} \sim \text{SHAPE}(\cdot|\mathbf{s}'_l). \tag{5.4}$$

Then the mask is re-sampled as described for the masked RBM in section 4.3.3 of chapter 4. Exact inference in the model given in equation (5.2) would mandate

$$P(m_i = l|\mathbf{h}'^{(s)}_{1..L},\pi') \propto \text{SHAPE}(s_{l,r_l(i)}=1|\mathbf{h}_l'^{(s)}) \prod_{l'\in o(i):\pi(l')<\pi(l)} \text{SHAPE}(s_{l',r_{l'}(i)}=0|\mathbf{h}_{l'}'^{(s)}), \tag{5.5}$$

where the product in the second term is over all patches that overlap with image pixel i and that are in front of patch l (note that, as in equation (5.2), we need to apply the

[4]Note that updating the relative depth of a single patch wrt. to its neighbors will only change the latent shapes of a subset of patches, those which overlap with the patch in question.

appropriate re-mapping from image pixels i to patch pixel $r_l(i)$). The first term in this expression corresponds to the probability of shape l turning on, the second term gives the probability that the shapes of all patches that are in front of patch l are off.

This, however, couples the shapes of all patches overlapping with a particular image pixel which is undesirable.[5] We therefore apply the same update as for the masked RBM (equation 4.21), which is consistent with the way we choose to (approximately) generate from the model:

$$P(m_i = l|\mathbf{h}_{1..L}^{\prime(s)}, \pi') = \begin{cases} \prod_{l' \in o(i): \pi(l') < \pi(l)} \text{SHAPE}(s_{l', r_{l'}(i)} = 0|\mathbf{h}_{l'}^{\prime(s)}) & \text{if } l \text{ is rear-most patch at pixel } i \\ \text{SHAPE}(s_{l, r_l(i)} = 1|\mathbf{h}_l^{\prime(s)}) \times \prod_{l' \in o(i): \pi(l') < \pi(l)} \text{SHAPE}(s_{l', r_{l'}(i)} = 0|\mathbf{h}_{l'}^{\prime(s)}) & \text{otherwise,} \end{cases} \tag{5.6}$$

This assumes that it is always the rear-most patch that is visible if none of the patches that are in front of that patch are turned on, independently of the preferences of its shape model.

5.1.4 Learning

Given a set of mask images $\mathbf{m}^{(1)} \ldots \mathbf{m}^{(T)}$ we would like to maximize the likelihood of the mask images under the occlusion model. Unfortunately it is not possible to perform this maximization directly and we are faced with effectively the same problems as in the case of the masked RBM (cf. section 4.3.3.2): It is neither possible to perform the summation over the unobserved latent shapes $\mathbf{s}_{l \ldots L}$ and depth ordering π, nor can the normalization constant of the binary shape RBM be computed. We therefore take an approach very similar to learning for the masked RBM: First, we (approximately) sample the latent shapes and the depth ordering from the posterior $P(\mathbf{s}_{1 \ldots L}^{(t)}, \pi^{(t)}|\mathbf{m}^{(t)})$ using the scheme outlined in the previous section. Secondly, we use these samples from the posterior as training data to compute a contrastive divergence update step of the parameters of the binary shape RBM as explained in section 4.3.3.2 (cf. equation 4.32). Note that this assumes that shapes are marginally independent and as explained

[5] This happens through the normalization (note the proportionality sign): If, for instance, the rear-most shape has a very low probability of turning on, the probability of the other shapes turning on increases.

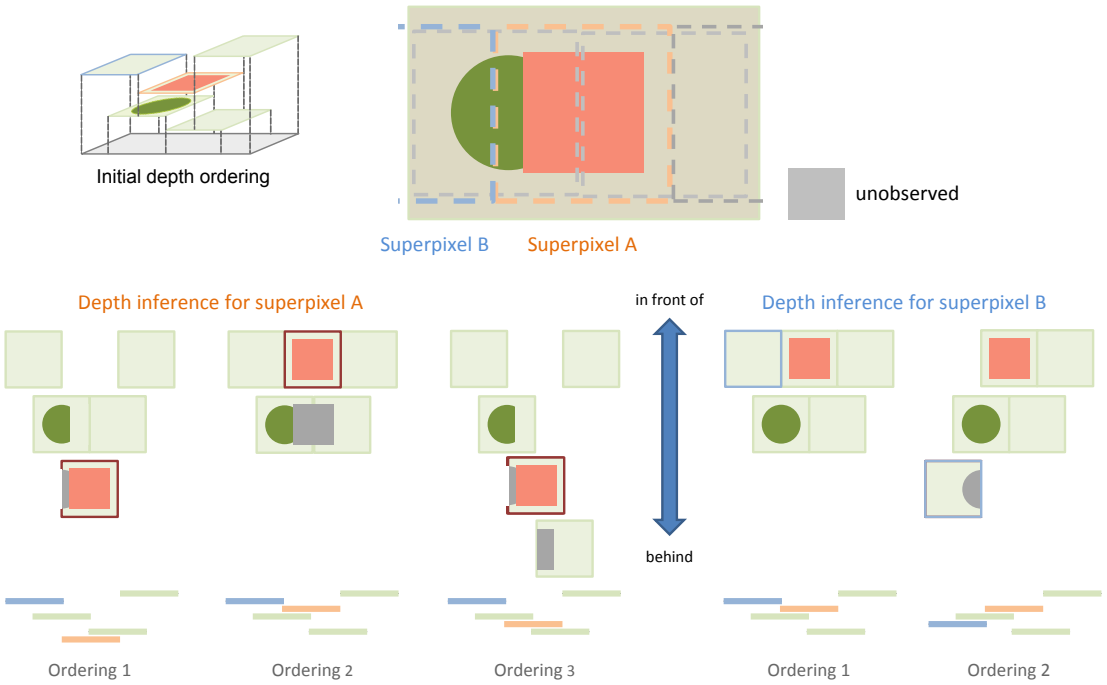


Figure 5.4: **Inference in the occlusion mask model for the field of masked RBMs** *Top*: Mask image with square-shaped and circle-shaped region. The latent patches governed by individual patch models are indicated by dashed lines. *Bottom*: Illustration of depth inference for the two highlighted patches ("superpixels") in the top panel. Given an initial depth ordering (inset in upper left corner) one superpixel is considered at a time (here, first superpixel "A" and then "B"). For the superpixel under consideration, the different depths relative to the position of its neighbors are explored. Superpixel "A" has two partially overlapping neighbors, so there are three orderings to be explored. Superpixel "B" only has one overlapping neighbor in this illustration, thus only two orderings need to be explored here. Each relative depth gives rise to observed and unobserved (shown in gray) pixels in the patch under consideration and also for the neighboring superpixels. The underlying idea is exactly the same as explained in Fig. 4.6 for the masked RBM.

in section 5.1.2, unlike in the masked RBM, learning is therefore only approximate[6].

[6]Exact learning in the model defined by equation 5.2 (in which shapes are not truly marginally independent) would require sampling full mask images by sampling all shapes jointly for computing the negative part of the gradient.

5.1.5 Integrating the shape prior with the appearance model

The joint distribution over the image and the latent appearances conditional on a mask defined by the FoMRBM is given by equation (5.1) (see section 5.1.1). This is the equivalent of equation (4.1) for the masked RBM (cf. section 4.2 in chapter 4) with the only difference that the image pixels need to be mapped onto the corresponding pixels of latent patches. Combining equation 5.1 with the occlusion-based mask model defined in the previous sections, the full model is obtained in the same way as for the masked RBM in section 4.3.4 of chapter 4.

5.2 Related Work

The FoMRBM is an extension of the masked RBM. Thus, some work relevant to the FoMRBM has already been covered in section 4.4. In particular, the work on generative models of natural image patches and the work on layered image models which we have discussed above (sections 4.4.1 and 4.4.2) is directly related to the FoMRBM as well. These two lines of work will be reviewed here only briefly. In particular, for the models of natural image patches there is not much to add to the discussion on their relation to the masked RBM in section 4.4.1 except that the FoMRBM is specifically designed to efficiently model larger images. Layered image models (cf. section 4.4.2) have typically been applied to larger images (like the FoMRBM) and they also allow placing objects in an occluding manner at different positions in the image. Yet, in the case of layered models, each image is represented in terms of only a small number of objects. Each object appears only once per image, but objects typically have associated transformation variables so that they can e.g. appear at different positions in different images. In the FoMRBM the objects are of limited size (size of the latent patch) and the positions where objects can be instantiated are fixed (the positions of the latent patches). On the other hand each latent patch has the potential to instantiate a large number of different objects (although it instantiates only one at any given time), and, in principle, the FoMRBM allows up to L (number of latent patches) object instances in any given image.

In the remainder of this section we will first discuss a connection between the FoMRBM and a popular notion in the computer vision literature, the “superpixel”. We will then focus primarily on generative models of images that are suitable for modeling larger images (in contrast to models of small image patches). We will further focus

on models that attempt to model images at the pixel level, i.e. models that are in principle applicable to tasks such as image segmentation or image editing. This excludes many approaches that define a generative model over feature maps extracted from images and which are typically designed to perform discriminative tasks such as object recognition. Some models that are directly related to the hierarchical extension of the FoMRBM will be discussed in Future Work in section 5.4.

5.2.1 Superpixel representations in computer vision

For many tasks, a pixel-level representation of an image can be problematic due to its high dimensionality. Furthermore it is highly redundant and to some extent arbitrary since it is effectively the result of the digital imaging process. Several recent works in the computer vision literature have therefore used a *superpixel representation* as the basis for further modeling or processing of an image, e.g. for segmentation (Ren and Malik, 2003; He et al., 2006), classification (Campbell et al., 1997; Mori et al., 2004), or for generating simple 3D reconstructions from single photographs (Hoiem et al., 2005). A superpixel representation is an over-segmentation of an image into coherent, and typically largely homogeneous regions that preserves most of the structure necessary for further processing. It can be obtained, for instance, using normalized cuts as e.g. in Ren and Malik (2003), and individual superpixels can then be described in terms of more abstract (usually manually defined) features reflecting their appearance and shape. In the case of segmentation, for instance, similar superpixels will be grouped together into larger regions forming the final segmentation of the image: Ren and Malik (2003) train a classifier that distinguishes good segmentations (i.e. groupings of superpixels) from bad ones based on Gestalt principles.

In the FoMRBM the mask associates image pixels with individual latent patches. Thus the field of masked RBMs learns to represent an image as a number of small regions each of which is explained in terms of an appearance and a shape. These regions can be thought of as *superpixels* although they differ from the standard notion of superpixels in that they are not required to be contiguous but merely constrained to lie within the boundaries of a latent patch. Such non-contiguity makes particular sense when dealing with occlusion, since the same superpixel can be used to represent parts of an object on either side of a narrow occlusion (see Fig. 5.12 for an example; due to this non-contiguity there can be more than L, i.e. more than the number of latent patches, regions in an image generated from the FoMRBM). Also, FoMRBM super-

pixels have high-order shape priors that have the potential to capture complex shapes, such as digits or letters and homogeneity is defined in terms of the appearance model. The interpretation of the FoMRBM as a superpixel algorithm will become clearer in the experiments described below. In some cases we will use the terms *superpixel* instead of *latent patch* in particular, when we refer to the state of a latent patch given an image (i.e. the image pixels that it is associated with, its latent shape and appearance).

5.2.2 Markov Random Fields as Image Priors

Probably the largest class of generative models of low and mid-level image structure that have been applied to images of realistic size is formed by MRF models (see also section 2.2.1.1). They have been used as priors over label fields for image segmentation, and the FoMRBM is, for instance, related to the "double MRFs" discussed in section 3.5.1 of chapter 3 (e.g. Derin and Elliott, 1987; Manjunath et al., 1990; Zhang et al., 1994; Melas and Wilson, 2002). Similar to the FoMRBM, these models define a region-based prior over images, although the focus of these works is more on applications such as segmentations than on learning a generative model of general natural image structure. The nature of these formulations is also different from the FoMRBM: in the FoMRBM an image is composed from many small regions that arise from the occlusion of many small independent "objects" or "parts" each of which has an associated relative depth. In the "double MRFs", however, region shape is typically determined by an MRF prior over the label field (e.g. a Potts model, cf. section 2.2.1.1 in chapter 2) , and the region appearance by some suitable appearance model, such as a Gaussian MRF. This allows for regions of arbitrary size at the cost of a less flexible prior of region shape and appearance (as well as typically fewer regions and no notion of depth).

Other MRF-based works aim more directly at modeling the statistics of natural images. Here, the emphasis is put on priors in image processing tasks, in particular for denoising or small-scale inpainting. Important examples include the Field-of-Experts model (Roth and Black, 2005) discussed in detail in chapter 3, the closely related FRAME model proposed by Zhu and Mumford (1997), and the seminal work by Geman and Geman (1984). The first two models are high-order MRFs with clique potentials defined in terms of the responses of linear filters. Geman and Geman (1984) propose a hierarchical model in which the four-connected neighborhoods of the lower layer are modulated by a higher-level line-process that can sever the (direct) depen-

dencies between two neighboring pixels. Two further examples are the works by Freeman et al. (2000) and Fitzgibbon et al. (2003) where clique potentials are quasi non-parametric in nature and defined in terms of the minimum Euclidean distance between the corresponding image patch and image patches in a large database extracted from natural images.

MRF-based image priors form a rather heterogeneous class of models. Nevertheless, there are some important general points to be made regarding the differences between these models and the FoMRBM. One aspect is that all the models referred to above are homogeneous MRFs, i.e. they are products of clique potentials that are replicated at all pixels across the image and they are thus truly translation invariant. This is different from the FoMRBM which only replicates latent patches on a coarser grid. Secondly, the interactions between the clique potentials in MRF models are very different from the interactions between the replicated latent patches: In the FoMRBM the latent patches compete to explain a pixel in a mixture-formulation. In contrast, in MRFs they interact in a product-of-expert manner. (Pixels *within* the same latent patch in the FoMRBM are modeled by a RBM, i.e. by a PoE.) Since the latent patches are of limited size and marginally independent the maximum range over which dependencies can be modeled in the FoMRBM is limited to the size of latent patches. In contrast, in MRFs dependencies typically extend well beyond the size of a clique. A final important difference arises from the nature of the latent representation that is obtained: The FoMRBM models an image directly in terms of small individual regions endowed with a shape and an appearance prior; the inferred shape and appearance are reflected in the state of the sets of hidden units. In contrast, MRFs typically do not provide a representation that directly reflects more abstract properties of the structure present in an image (but see Geman and Geman (1984) where the line process can be thought of as a representation of edges / region boundaries).

5.2.3 Image models from the Deep Learning Literature

Most work in the deep learning literature has focused on relatively small image patches. Nevertheless, some models suitable for larger images have been explored, in particular in the form of convolutional RBMs (Lee et al., 2009; Desjardins and Bengio, 2008; see also discussion in section 2.2.4.3 in chapter 2). Lee et al. (2009) extend the standard convolutional RBM to a multi-layer, hierarchical model, and introduce probabilistic pooling layers inspired by Convolutional Neural Networks (LeCun, 1989b,a).

It might seem as if the convolutional RBM and the FoMRBM have some similarities, but the differences are the same as for the MRFs discussed in the previous section: Firstly, the FoMRBM as described above is currently not fully convolutional. Secondly, and more imortantly, a RBM is a *product* of experts[7], whereas the FoMRBM is a mixture. The overlapping latent patches are marginally independent of each other, and only one of the latent patches explains each pixel. In contrast, in the convolutional RBM the hidden units with partially overlapping receptive fields are not independent but interact to form the image.

Recently, Ranzato et al. (2010b) have investigated several undirected models with latent variables, suitable for modeling full images. These are collectively referred to as "gated MRFs" and include extensions of the mcRBM (Ranzato and Hinton, 2010), the PoT (Teh et al., 2003), and the hierarchical PoT model (Osindero et al., 2006). The authors propose a "tiled convolutional" architecture in which cliques are not replicated at each image pixel but rather overlap in a manner very similar to the overlap of latent patches in the FoMRBM[8]. This similarity with the FoMRBM is, however, rather superficial since gated MRFs remain PoE (FoE) models and the hidden units with partially overlapping receptive fields interact in a manner very different from the hidden units of the latent patches in the FoMRBM.

Nevertheless, as discussed for the mcRBM (cf. section of chapter 4.4.1), there is some conceptual similarity: Gated MRFs explicitly model the covariance between image pixels. This allows these models to account for edges and region boundaries by modulating the correlations between image pixels. In spirit, this is related to the FoMRBM which modulates the correlations between image pixels in a hard manner by explaining them either with the same (image pixels strongly correlated) or different superpixels (image pixels independent). Unlike in the FoMRBM there are, however, no truly independent regions, and the decomposition of the image is not made explicit in the representation (there is no separation of region shape and appearance, and there is no notion of depth). An important advantage of the gated MRFs for images over the FoMRBM is that they can potentially model long-range correlations even with a limited receptive field size whereas in the FoMRBM the dependencies are limited to the extent of a latent patch. Furthermore, gated MRFs can achieve a soft-partitioning

[7] In fact, after summing out the hidden variables the convolutional RBM can be thought of as homogeneous MRF (cf. section 2.2.4.3 in chapter 2).

[8] In their formulation, the receptive fields are laid out in several grids. The hidden units whose receptive fields make up a single grid have non-overlapping (abutting) receptive fields and share weights, as is the case in the FoMRBM, but the weights are not shared across layers.

of the image whereas the partitioning obtained with the FoMRBM is always hard.

5.2.4 Other generative image models

Three other models shall be discussed in their relation to the FoMRBM. Firstly, as already pointed out in relation to the masked RBM, the idea that occlusions are an important property of the natural image formation process can be found at various places in the literature (cf. section 4.4). For instance, Ruderman (1997) makes a connection between scale invariance of many image statistics and the fact that 2D images arise from typically a large number of independent, occluding objects. The idea of randomly overlapping objects has inspired simple models of image clutter (Grenander and Srivastava, 2001; although this model does not implement an actual occlusion-like nonlinearity). Models which assume that images are formed from simple, template-based elementary "objects" (e.g. circles, squares, etc.) that (partially) occlude each other and whose positions and possibly scale are randomly drawn from a Poisson process are referred to as "dead leaves models". Such a model and its ability to reproduce basic image statistics has been studied e.g. in Lee et al. (2001) (see also Jeulin, 1997) who found a good quantitative fit between the artificial images generated from the model and natural images for the statistics considered, despite the conceptual simplicity of the model. There are several similarities between the FoMRBM and the dead leaves model. In particular, the fact that images are formed from a potentially large number of objects, that individual objects are independent of each other, and the idea that important image properties arise from the fact that these independent objects occlude each other. However, there are also crucial differences: (i) The dead leaves model considered by Lee et al. uses only a single, very simple template shape such as a square or a circle (in some cases with a transformation parameter to allow for rotations); (ii) there is an, in principle, infinite number of objects for each image; (iii) there is no inherent constraint on object parameters: locations and transformation parameters are drawn from appropriate distributions that shape the statistics of the generated images. In the FoMRBM the maximum number of objects is fixed and there are constraints on the positions and the size of the individual objects. At the same time, instead of using a single shape template with a homogeneous color, the potential set of shapes and appearances is very rich as these are modeled by RBMs.

Secondly, (Guo et al., 2003, see also section 2.2.6 in chapter 2) propose a generative model of cluttered, texture-like images that also bears some resemblance to the

FoMRBM. The model composes images from several layers of textons (which in Guo et al., 2003 are implemented as partially transparent templates that can be translated, scaled, stretched, and rotated). The positions of the individual textons in each layer are modeled by a MRF which constrains the relative position and alignment of neighboring textons. The textons in different layers occlude each other. The authors describe a way to learn the model from images (fitting a different model for each texture). Similar to the dead leaves model and unlike the FoMRBM this model does not fix a-priori the number of textons in each layer (or in the full image). Furthermore, it constrains the relative arrangement of textons in each layer via a MRF so that individual textons are not independent, unlike the latent patches in the FoMRBM. On the other hand, the model is simpler than the FoMRBM in that the experiments presented in Guo et al. (2003) use only a single template per texton layer whereas the FoMRBM uses potentially very rich shape and appearance priors for the latent patches. Furthermore, a depth ordering is only defined between layers, but not between individual textons within a layer (note that the term layer is used rather differently by Guo et al. than it is in the FoMRBM: Guo et al. group textons into layers, all texton instances in the same layer are generated from the same template and are at the same depth; textons instances in different layers use different templates, are governed by independent MRFs, and occlude each other).

Finally, Kannan et al. (2007) propose a generative model that composes an image from "jigsaw pieces" that are taken from a jigsaw image (typically smaller than the original image). In this model, a jigsaw piece is simply a set of coherent pixels in the jigsaw image. Pieces can have arbitrary shapes but the energy of the model encourages the use of large, coherent pieces. Jigsaw pieces are placed in the image to be generated in an abutting manner, i.e. the image is effectively generated by copying from the jigsaw image and pasting (in a non-overlapping manner) into the target image. When a jigsaw is learned for a given image (or set of images) the jigsaw image typically ends up containing parts that recur in the dataset. When trained on face images, for instance, the learned jigsaw contains "parts" of faces, i.e. groups of pixels that look like eyes, noses, etc.. The model is similar to the FoMRBM in that it pieces together an image from a large number of small parts, although unlike in the FoMRBM there is no restriction on the size of a jigsaw piece (a piece is simply defined by its coherence in the jigsaw image). One important difference is that the shape of these jigsaw pieces is not modeled – unlike the shape of latent patches. Thus, even though certain coherent groupings of pixels in the jigsaw image might be especially likely in the training data

(e.g. pixels that jointly look like an eye) there is nothing in the model that encourages generated images to be composed from those groupings as compared to other equivalent groupings. Thus, the model is suitable for segmentation, but not necessarily for image generation. Given the lack of a notion of shape there is also no explicit notion of occlusion in the model (although occlusion boundaries can potentially be recovered by identifying rare transitions).

5.3 Experiments

We evaluated the FoMRBM with the occlusion shape model on two datasets. In the first set of experiments described in section 5.3.1 we applied the model to a set of generated images containing randomly overlapping shapes. The purpose of this first set of experiments was to demonstrate the general viability of the model and to illustrate some of its fundamental properties. In the second set of experiments we trained a FoMRBM on natural images. These experiments served to demonstrate that the model can learn a reasonable generative model of more complex data. In particular, these experiments illustrate the superpixel representation discussed above (section 5.2.1) and they show how the model can be applied to simple image editing tasks such as inpainting.

5.3.1 Experiments on Toy Data

In this section we describe a set of experiments in which we train the model on artificial images generated to contain a small number of randomly overlapping shapes. The experiments serve to demonstrate that the model can recover the ground truth shapes from cluttered images and that the model performs inference correctly despite the various approximations.

5.3.1.1 Dataset

The toy dataset consisted of 100 RGB images of size 80×80 pixels in which "objects" of 5 different shapes randomly occluded each other. The images were generated using the appearance model also used for the experiments with natural image patches above (cf. section 4.5.2) and five binary shape templates. Shapes were laid out to be consistent with the layout of latent patches described for $K = 4$ in section 5.1.1: First, one set of latent patches (corresponding to one of the four grids of non-overlapping patches shown in Fig. 5.2) was chosen to form the background and the corresponding latent

patches were filled with a particular color. The remaining latent patches were then chosen randomly either to be off or to be "filled" with one of the five shape templates and with one of 7 colors (different from the background color). Shape colors were chosen such that overlapping shapes would not have the same color. In all cases the actual colors used to fill background and shape latent patches were obtained as samples from the appearance model. Some example images are shown in Fig. 5.5.

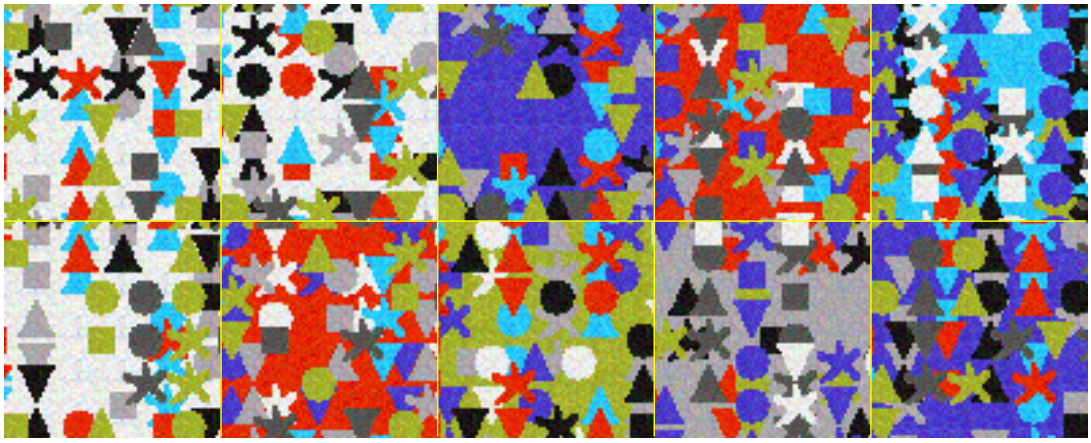


Figure 5.5: **10 examples of training RGB images for the field of masked RBM**. There are five different shapes and five different colors. No two overlapping shapes have the same color.

5.3.1.2 Training

We trained a field of masked RBMs with latent patches laid out as explained in section 5.1.1 above for $K = 4$. Each of the grids consisted of 6×6 latent patches (i.e. $4 \times 6 \times 6 = 144$ latent patches in total for a full image), and the grids were offset relative to each other by 8 pixels horizontally and vertically (none of the grids was aligned with the image boundaries). The Beta RBM used to generate the training images was used as appearance RBM. The shape RBM was chosen to be a binary RBM with 20 hidden units. The weights were initialized randomly.

The dataset was split into 100 "mini-batches", i.e. each training image was processed separately. Depth ordering, latent shapes, and appearances were initialized randomly for each image. Masks were initialized so that each latent patch was associated with the image pixels corresponding to the 8×8 pixel square in its center. Training was performed for 100 epochs. In each epoch we performed 5 iterations of inference for each image, each iteration involving a full update of the latent appearances, of the

depth ordering, of the latent shapes, and of the mask. The shapes inferred for the latent patches were then used as training data to update the binary shape RBM as described in section 5.1.4. Latent patches at the image boundary that overlapped with the image to less than 25% were not included in the update, similarly, the shapes of latent patches that had been inferred to be behind all other patches were excluded. Learning parameters where chosen as follows: We used CD-5, the learning rate was set to 0.03 and the momentum to 0.5.

5.3.1.3 Results

We assessed the quality of the learned shape model in two ways: Firstly, we sampled from the binary RBM to verify that it had indeed learned about the shapes prevalent in the training images. Secondly we used the full model to perform inference on shape images to assess whether the model would be able to (a) segment the images correctly, and (b) to infer the correct relative depths of overlapping patches.

Samples from the binary shape RBM were obtained by randomly initializing the hidden units and performing Gibbs sampling for 10000 iterations. Results are shown in Figure 5.6. These samples suggest that the model has indeed learned about the shapes that the test images were composed from: it generates predominantly valid and complete shapes despite the strong presence of occlusions in the training data. The samples do not reflect the relative frequency of shapes in the training images (for instance, the triangles are sampled less frequently than the other shapes), however, this is less likely to be an indication of a general failure of the model than of the fact that contrastive divergence training tends to have difficulties to estimate the relative mass of modes correctly (e.g. Hinton et al., 2003 and discussion in section 2.3.2 of chapter 2). Furthermore, not all of the samples correspond to valid shapes. This is expected since the training images only partially have to be explained in terms of shapes. As shown below, the background will typically be segmented in an unspecific manner which will be reflected in the shape model that is being learned. Note that the model is not necessarily expected to learn a special "all-on" background shape: In most areas of an image the representation is highly overcomplete, i.e. there are many more latent patches available than required to accurately model the image structure. The model can decompose the background into several, unspecific overlapping shapes, similar to the behavior of the MRBM for homogeneous image patches discussed in the previous chapter (section 4.6.1). This issue will be discussed further below.

Example inference results for one test image (the second image in Fig. 5.5) are

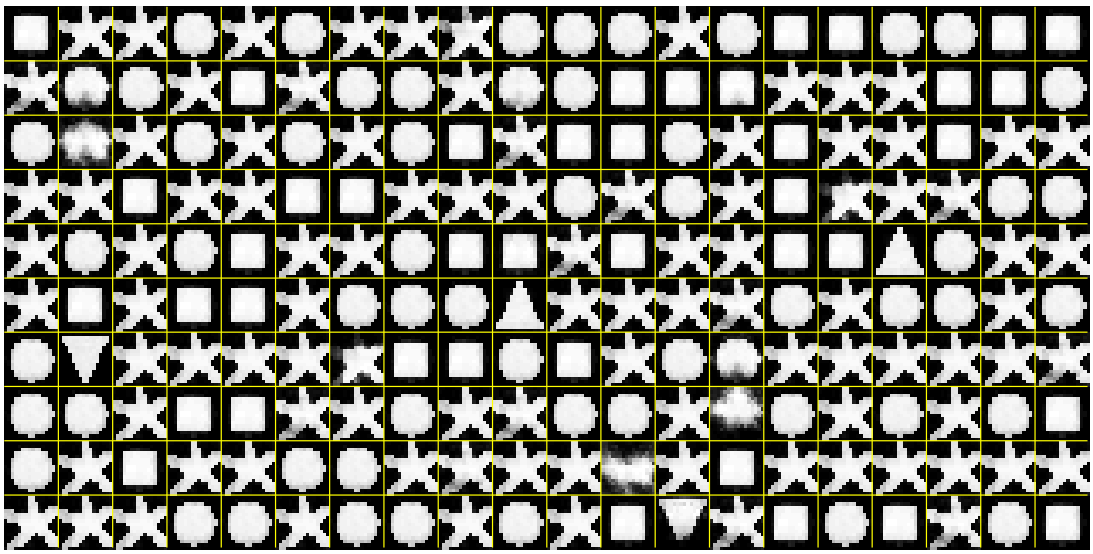


Figure 5.6: **Samples generated from the shape model** learnt using training images such as the one shown in Fig. 5.5, after running a Gibbs sampler for 10000 steps. The images shown are the probabilities of the binary visible units given the binary states of the hidden units. Though most of the shapes in the training data are partially occluded, the model learns to generate complete shapes.

shown in Figures 5.7 and 5.8. These inference results were obtained by initializing the depth ordering, the latent shapes, and appearances randomly. The mask was initialized such that each latent patch was allocated the 8×8 pixels in its center. Inference was then run for 100 iterations (re-inferring the relative depth of all patches, the latent shapes and appearances, and the mask in each iteration). Fig. 5.7 (left) shows the segmentation inferred by the FoMRBM. The yellow lines indicate the boundaries between regions explained by different latent patches (i.e. the image shows effectively the region boundaries in the mask image). The model isolates the individual shapes largely correctly. This is positive, although perhaps not too surprising considering the way the test images had been generated. More interesting are the results shown in Figure 5.7, right. Here, the segmentation has been augmented with the inferred relative depth between neighboring regions. Each segmentation boundary has been marked with a red-green double-line, the red-side of the boundary pointing toward the latent patch that has been inferred to be in front, and the green side of the boundary pointing toward the region that has been inferred to be in the rear. This figure suggests that the model not only segments the image correctly but that it is also able to infer the relative depth of the neighboring regions largely correctly. Figure 5.8 illustrates the driving

force behind depth inference: The Figure shows the complete latent representation inferred for the image: For each latent patch the inferred latent appearance masked by the inferred latent shape are shown. The inset illustrates, for three of the latent patches, how shape and appearance are combined to obtain the joint representation. The latent shapes in the grid are arranged according to their relative position in the image (for three latent patches the corresponding image regions are highlighted in Fig. 5.7 to serve as a reference). Although many shapes in the image are partially occluded the model has nevertheless inferred the true (unoccluded) shape and appearance in most cases. It is this ability of the model to "complete" occluded shapes that drives depth inference.

One property of the segmentation shown in Fig. 5.7 that deserves further explanation is the treatment of the background. The figure shows that the background is broken up in a relatively unstructured manner. This is the expected behavior since for the homogeneous background there is no evidence from the image that would contribute to the segmentation. The shape model will attempt to represent such regions largely in terms of overlapping shapes (all of the same appearance) but since there is no evidence to guide this segmentation it will remain uncertain leading to a noisy mask (and an unconfident depth inference). If additional cues are given, however, then such a "hallucinated segmentation" can become rather confident giving rise to phenomena similar to certain visual illusions. This is illustrated in Figure 5.9. This figure shows a simple version of the "Kanisza triangle" for the model at hand: Although there is no actual white triangle present in the image the model obtains a rather confident representation in terms of three intact circles occluded by a white triangle.

5.3.2 Experiments on natural images

The experiments on toy data in the previous section serve to demonstrate the general validity of the model and that it is indeed possible to perform inference and learning in the model. In this section we will apply the model to natural images. We will show that the model is able to learn a reasonable model of shapes present in natural images and that the FoMRBM with occlusion shape model gives rise to a reasonable model of mid-level structure in natural images that can be applied to tasks such as image segmentation and simple infilling problems.

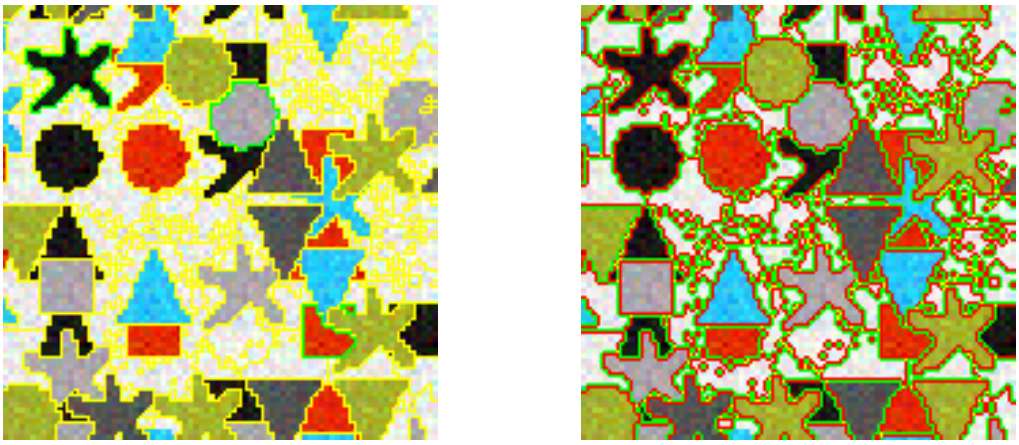


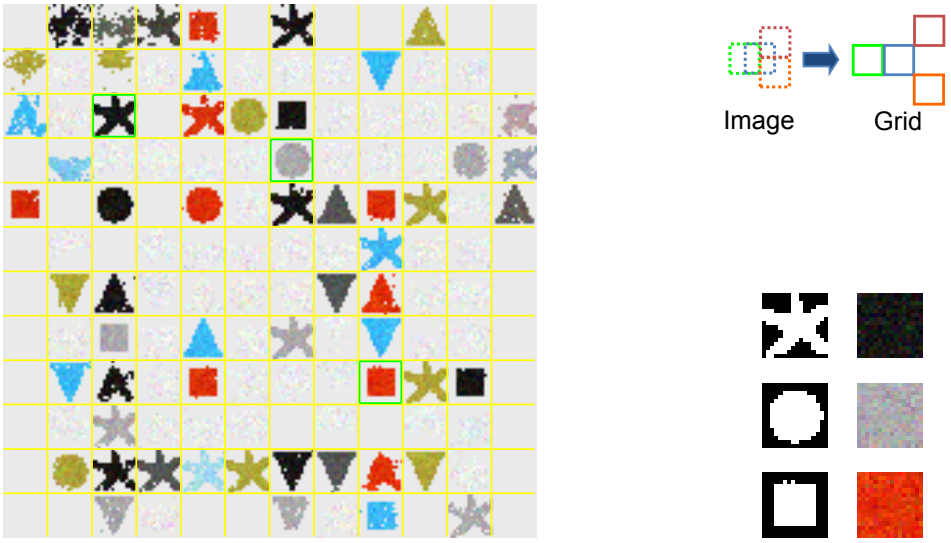
Figure 5.7: **Inferred segmentation and relative depth**. *Left*: Areas explained by different superpixels (latent patches) are delimited by yellow lines. The areas explained by three of the superpixels are highlighted in green. These serve as a reference to relate the segmentation image to the "exploded" view in Fig. 5.8. *Right*: Segmentation and inferred relative depth between neighboring superpixels. The segmentation boundaries are double-marked by red and green lines. The red line indicates the side of the boundary that has been inferred to be in front, the green line the side that has been inferred to be in the rear. In most cases in which two ore more shapes overlap in the image the model inferred the depth ordering correctly. Note that there are several cases where it fails to place shapes in front of the homogeneous background. This is due to the fact that the model infers an unspecific segmentation for the background making it harder to infer the relative depth.

5.3.2.1 Dataset

Our training set consisted of 1000 images of size 80×80 pixels extracted from RGB images downloaded from the web. Some example images are shown in Fig. 5.10. No pre-processing was applied to the images.

5.3.2.2 Details of model and training

For the field of masked RBM we chose the same layout of patches as in the experiments with the toy shape data in the previous section. Thus, each image was covered by 144 latent patches. The patches were of size 16×16 pixels and were laid out in $K = 4$ spatially offset grids, each of which consisted of 6×6 non-overlapping patches. The



(a) (b)

Figure 5.8: **Latent representation inferred for the test image**. The left panel *(a)* shows the latent appearance and shape for each of the 144 latent patches that jointly represent the image. Each cell in the grid corresponds to one latent patch and shows the appearance fantasy masked by the shape fantasy that has been inferred for the respective patch. Latent patches are arranged in the grid according to their position in the image. For instance, the black star in the 3rd cell (from the left) in the third row of the grid corresponds to the black star in the top left corner of the image. To serve as a reference, three superpixels that have been highlighted in green in this figure and also in Fig. 5.7. The illustration to the right (above panel *(b)*) shows how superpixels with partially overlapping bounding boxes in the image are laid out in the grid. The panel on the right *(b)* shows, for the three highlighted superpixels, the inferred shape and appearance fantasies separately. Note that that in most cases the model has successfully completed the sometimes heavily occluded shapes in the original image. It is this ability to complete shapes that drives depth inference.

grids were offset horizontally and / or vertically by 8 pixels.

For the appearance model we used the pre-trained Beta RBM which we already used for the experiments with the masked RBM (see section 4.5.2.1 for details). The binary RBM for the shape model was chosen to have 384 hidden units. The weights were randomly initialized at the beginning of learning.

The shape model was trained in two phases:

test image inferred segmentation latent representation

Figure 5.9: **Kanisza triangle**. *Left*: Test image. *Middle*: Inferred segmentation (yellow) and relative depth ordering for some of the superpixels. *Right*: Inferred latent representation (same format as in Fig. 5.8). This figure illustrates the model applied to a toy version of the Kanisza triangle. The model described above was used to perform inference for the image on the left. Even though there is no actual white triangle present in the image the model "hallucinates" such a shape: The three blue circles with missing corners are explained in terms of three full circles and an occluding white triangle. Given its knowledge about triangles, squares, circles, and stars the inferred representation is more plausible than the representation in terms of circles with corners cut out.

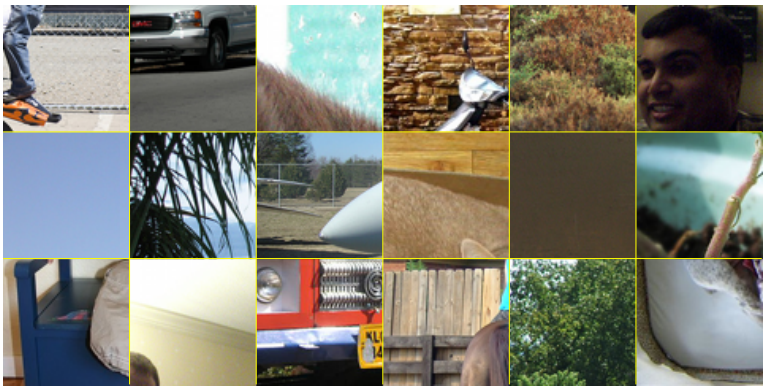


Figure 5.10: Examples of the 80×80 pixels images used to training the shape model for the field of masked RBMs

1. For pre-training we inferred the mask for natural images of size 80×80 pixels (RGB) with a field of masked RBMs, using the uniform model as mask prior (cf. eq. 5.1), and running 100 iterations of mask inference. From each image we obtained 144 binary shape patches (i.e. the binary shapes associated with each of

the 144 latent patches covering the image). We randomly selected 95000 binary patches (excluding any mask patches from superpixels not fully overlapping with the images) and used those as training data for a binary RBM (256 visible units, 384 hidden units). Training was performed for 10000 epochs using stochastic maximum likelihood (Tieleman, 2008), using a learning rate of 0.0005, no momentum, a weight decay of 0.0002 and mini batches of size 100. The parameters of this binary RBM were used to initialize the shape model for training in the context of the full FoMRBM.

2. We subsequently trained the occlusion-based shape model in the context of a field of masked RBMs, initializing the binary RBM for the shape model with the parameters obtained in phase 1. "Batches" consisted of individual images (note that there are 144 superpixels associated with each image). As described in section 5.1.4 we alternated inference and the update of the model parameters. Two iterations of full inference (update of the appearance fantasies, shape fantasies, relative depth for all superpixels as well as of the mask) were performed for each image before computing the gradient and updating the parameters. As explained in section 5.1.4 inference in the mask model was performed in parallel for patches which did not share neighbors (i.e. for patches that were independent conditioned on the mask and the remaining, non-overlapping patches) and such sets of independent patches were treated sequentially but in a random order. 10 steps of masked Gibbs sampling were performed to update the shape fantasies. Completely unobserved superpixels were forced to be in front, i.e. their shape fantasies were required to be completely off, in order to prevent unconstrained hallucinations by the model. We used CD-15 for training, using a learning rate of 0.0025, weight decay of 0.0002, and momentum of 0.5. To prevent the model from learning from largely unconstrained shapes (its own hallucinations) we did not include shapes from superpixels into the gradient that overlapped with the image to less than 25%. Training was run for 100 iterations and took approximately three weeks using our unoptimized Matlab implementation on a single-core machine.

5.3.2.3 Results

To evaluate the model and to illustrate its properties we performed three experiments on the trained model:

Figure 5.11: **Test image for inference with the FoMRBM** (320×213 pixels)

- Inference: We applied the full trained FoMRBM to test images and assessed the plausibility of the inferred segmentation and depth ordering.
- Image editing: We applied the model to simple infilling tasks, where we required the model to fill in small areas of pixels that were removed from the original image.
- Sampling: We generated samples (full images) from the field of masked RBMs

Inference: We used the the trained full model to perform inference on the image shown in Figure 5.11. The results are shown in Figure 5.12 which shows the equivalent of Figs. 5.7 and 5.8 for the toy data. It shows the segmentation inferred by a field of masked RBMs with the occlusion shape model together with the corresponding latent representation of all 1218 patch models (of size 16×16 pixels) covering the image of size 320×213. For each superpixel, the *combined* latent shape and appearance are shown (as in Fig. 5.8). For the toy data considered in the previous section, the model was confident with respect to the shapes composing the image and with respect to their relative depth. In contrast, for real images such as the one considered in Fig. 5.12, there is considerably more uncertainty as to what a suitable decomposition of the image would be. Not only are relevant regions typically significantly larger than the extent of the individual patch model, but there is also an enormous variability of shapes in natural images. With only the very local information available to the model, a decomposition in terms of high-level components of the scene cannot necessarily be

expected. Nevertheless, the decomposition of the image that is inferred by the model appears largely sensible: in particular, it has a tendency to explain the image in terms of small shapes, especially thin horizontal and vertical structures, that appear in front of larger homogeneous backgrounds. This is very noticeable when focusing, for instance, on the representation of the various signs in the image ("Except for access", "al Service", "TY Ltd", and the "no parking" sign) where the letters have largely been separated out and are placed in front of mostly contiguous background superpixels[9]. Note that, due to the explicit representation of occlusions, superpixels in the rear do not have to model the cut-out shape of foreground superpixels (even though there are some counter-examples, e.g. the "x" and "c" in "Except" are being explained in terms of a black background of unspecific shape behind a light gray foreground that has the letter shape cut out). Other examples are the frames of signs and windows which have predominantly been explained in terms of thin horizontal and vertical structures with often larger superpixels in the rear. To facilitate the mapping between the two representations, we have color coded superpixels in both sub-figures, representing letters in red, superpixels representing the background of the signs in blue and some of the superpixels explaining window frames in green.

The nature of this decomposition is the result of training the field of masked RBMs on a large dataset of natural images. Many of the training images images are efficiently explained in terms of thin structures in front of larger "background" patches. Furthermore, thin horizontal and vertical structures are especially frequent in natural images, and accordingly the models preference for separating these into "foreground" patches is particularly robust.

[9]To avoid any misunderstanding: We are not suggesting that the model has actually learned about letters. The letters are merely examples of the kind of structure the model like to place in the foreground

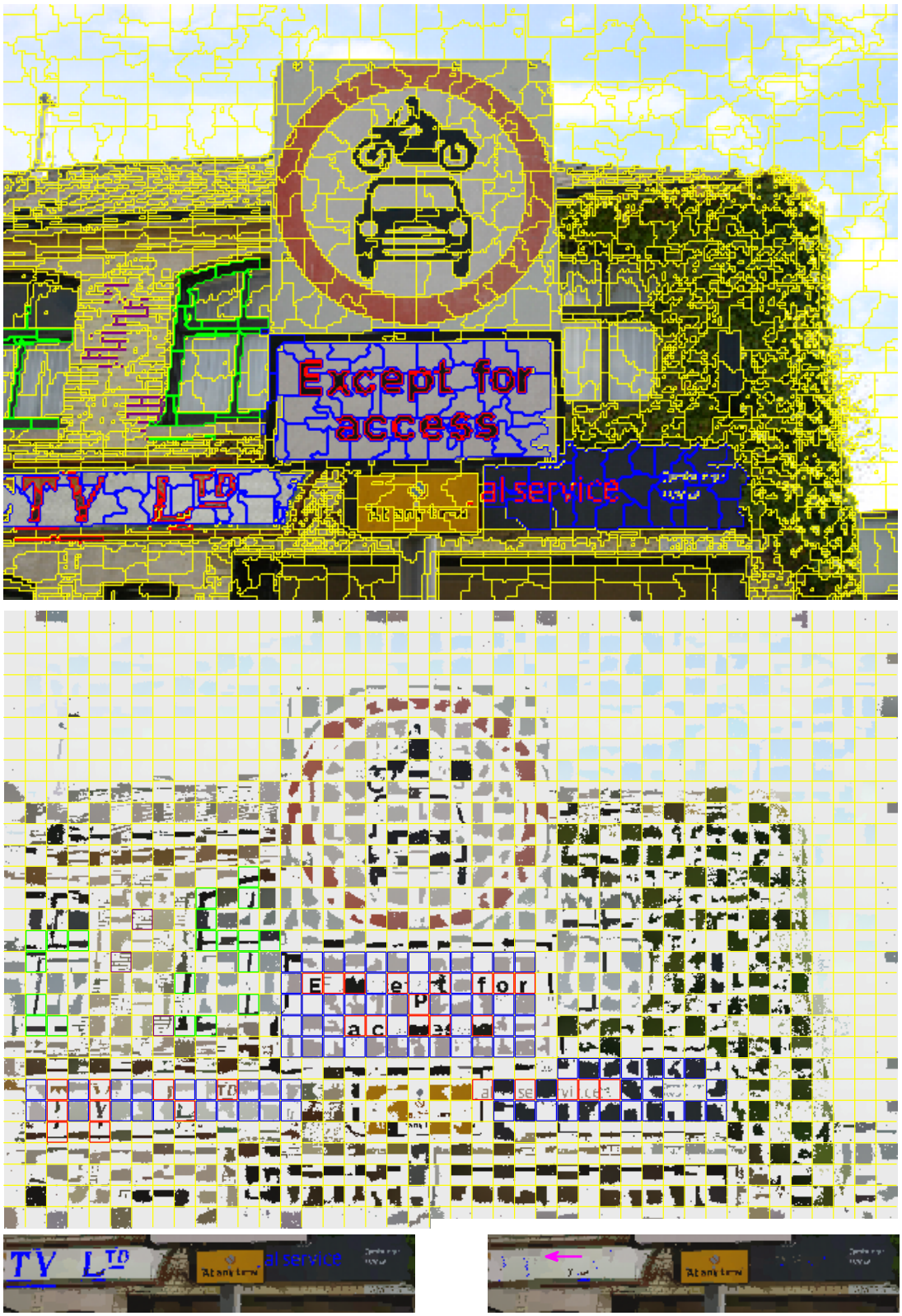

Except for
access
TV LTD
al service

Figure 5.12: Inferred segmentation and latent representation of superpixels

Top: inferred segmentation of the test image shown in Fig. 5.11. Areas explained by different superpixels are separated by yellow lines. Some superpixels are color coded as explained below. *Middle*: inferred latent representation of all superpixels covering the image (there are 1218 partially overlapping superpixels of size 16×16 pixels laid out as explained in Fig 5.2). Each grid cell corresponds to one superpixel. Superpixels are arranged according to their relative position in the image. Note that superpixels that are horizontally or vertically adjacent in the grid overlap in the image by 16×8 and 8×16 pixels respectively, while diagonally adjacent superpixels overlap by 8×8 pixels, as shown in the close up in Fig. 5.2. Superpixels that are separated by one cell in the grid are adjacent (non-overlapping) in the image. For each superpixel, the inferred latent appearance is masked by the inferred latent shape as for the toy data in Fig. 5.7. The model has a tendency to explain the image in terms of small shapes, especially thin horizontal and vertical structures, that appear in front of larger homogeneous backgrounds. This is reflected e.g. by the latent representation of the various signs in the image, for which the letters have largely been separated out correctly (superpixels color-coded in red) and are inferred to be in front of mostly contiguous background superpixels (color-coded in blue), and also in the representation of the frames of signs and windows which have predominantly been explained in terms of thin horizontal and vertical structures (some superpixels are color-coded in green) with typically larger superpixels in the rear. Note also how the set of non-contiguous image regions corresponding to the bricks in the wall is represented by superimposing a fine grid of mortar (superpixels colored in purple) onto a small number of larger brick-colored (latent) shapes. *Bottom*: reconstruction of part of the image from the inferred mask (*left*) and the inferred latent shapes (*right*). To illustrate that "background" superpixels are filled in underneath the foreground structure, we reconstruct the image using the inferred appearances, shapes, and depths for each superpixel, ignoring all superpixels corresponding to letters (color-coded in red in the top and middle panels). For the left hand figure, we use only the *visible* parts of the shapes of the superpixels (corresponding to the mask which is represented by the segmentation outline in the top panel). This means that pixels belonging to the letters are missing in this reconstruction (highlighted in blue). In the right-hand figure, we reconstruct the image using the inferred *latent* shapes as shown in the middle panel. These inferred shapes are larger than the visible parts of the superpixels so they partially occlude each other. Since the letters had been inferred to be in the foreground, the model was able to largely fill in the missing pixels in the background superpixels. Accordingly, many fewer pixels are missing in the reconstructions (there is some noise from sampling the unobserved parts of the shapes). Note also that some structure in the background arising from shading of the sign that is partially occluded by the letters has been completed in a meaningful manner (purple arrow).

Inpainting: To further illustrate the value of the shape model, we show the behavior of the model on a simple structure inpainting task in Fig. 5.13. In several places, image pixels overlapping with region boundaries were removed and treated as unobserved during inference (there are seven such "unobserved" areas with an average size of more than 26 pixels in the first example of Fig. 5.13, top left panel). The learned shape prior allows the model to continue region boundaries across the unobserved parts of the image, giving rise to a plausible reconstruction of the removed pixels (Fig. 5.13, middle panel). Inference is done by sampling and there is some uncertainty with respect to the correct reconstruction. This is reflected in the mean reconstruction (Fig. 5.13, right panel) for which some of the filled-in boundaries are slightly blurred. The model is however relatively confident in most cases. Note that the ability of the model to perform such a task crucially depends on the shape model.

Sampling: The field of masked RBMs defines a generative model of natural images and it is possible to draw samples from this model. Fig. 5.14 shows images of size 80×80 pixels generated from a field of masked RBMs trained on natural images. Samples are obtained by first sampling shapes and appearances independently for each of the 144 latent patches covering the image and then composing them according to a random depth order (as pointed out above, this generative process bears some resemblance to the "dead-leaves model" (Lee et al., 2001)).

The generated images contain many regions arising from partially overlapping 16×16 pixel square patches. This is to be expected considering that the training data contains large homogeneous regions which are well explained in terms of such almost completely filled superpixels (see also the discussion of the inferred latent representation in the previous section). In addition to that, the samples also contain many regions with smooth, non-rectangular boundaries that cannot be explained in this manner. These reflect the shapes of boundaries found in natural images which have been learned by the shape model. Individual samples from the shape model are shown in Fig. 5.15. These suggest that the shape model has developed a strong preference for coherent structures, and in particular for smooth horizontal and vertical boundaries. The model appears to have a preference for larger shapes but also generates some smaller structures. Another notable property of these samples is that the model exhibits a bias towards shapes that occupy the upper half of a patch. This bias presumably arises from the layout of patches across an image and from the fact that the representation is in most cases highly overcomplete (for each image pixel there are $K = 4$ latent patches that could explain that pixel) and thus the inferred segmentations highly undercon-

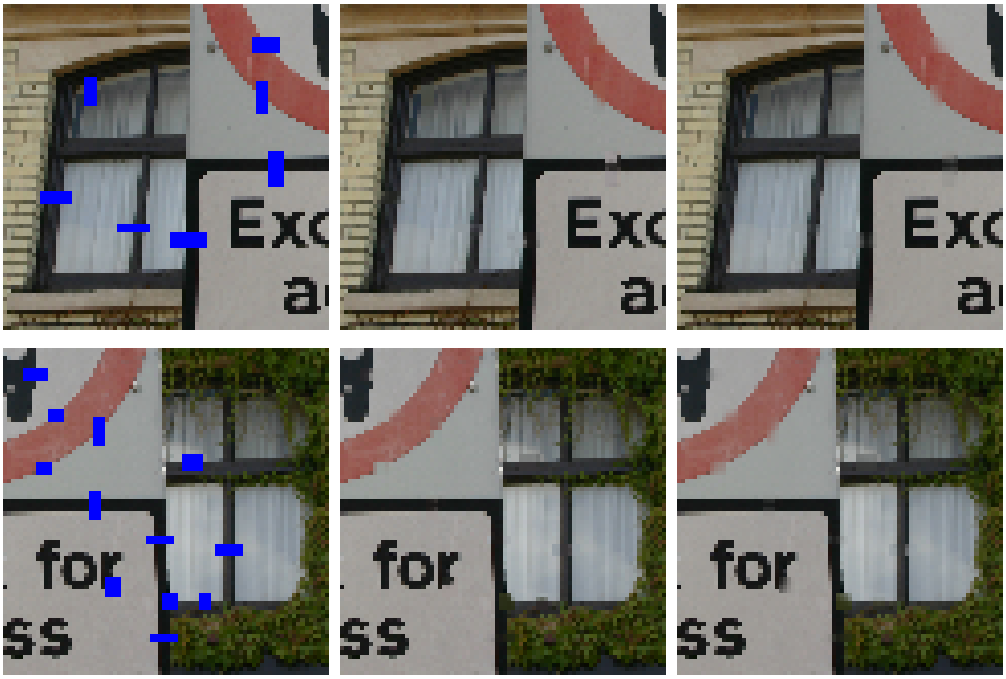


Figure 5.13: **Structure inpainting in images** The figure demonstrates the value of the shape model on a simple structure inpainting task which requires knowledge about the shape of boundaries in natural images. *Top, left*: input image, 80×80 pixels with seven regions of "unobserved" (missing) pixels (unobserved pixels are colored in blue; the average size of each region is more than 26 pixels). Regions of unobserved pixels were chosen so as to overlap with region boundaries in the image. Full inference was run on the input image for 200 iterations (inferring the mask as well as shape and appearance fantasies and the depth order; each iteration corresponds to one full update of all latent variables) treating pixels in the blue regions as unobserved. *Top, middle*: at the end of the inference, the unobserved pixels were filled in using the inferred latent shape and appearance fantasies. Note that the model fills in the unobserved parts of the image largely correctly, continuing the boundaries of image regions in a plausible manner. This relies on the shape model having acquired knowledge about plausible region shapes during learning. *Top, right*: pixel-wise average of the reconstructions obtained during the last hundred iterations of inference. Taking such average is not a good way of doing inpainting since it ignores correlations between neighboring pixels, but it is shown here to give some indication of uncertainty in the reconstructions (inference is done by Gibbs sampling, thus the inferred latent shapes, appearances, and depths can and will vary from one iteration to the next). *Bottom*: Results for a second example image. Same format as top row.

strained (e.g. for a fully homogeneous image $K = 1$ would be sufficient), as will be discussed in more detail below.

These characteristics of the samples (and also the nature of the latent representation inferred for real images discussed in the preceding section) suggest that the field of masked RBMs does indeed learn a sensible representation of natural images. At the same time, however, they also indicate one structural deficit of the model: individual patch models are assumed to be independent of each other. This is not necessarily a problem when performing inference since, in this case, the relevant longer-range dependencies are prescribed by the observed data (in fact, it helps keeping inference tractable). Yet, when generating from the model, this means that shapes and appearances of neighboring patches are not required to be consistent with each other, giving rise to the more or less random patchwork of shapes and appearances observed in Fig. 5.14. This makes it very unlikely that the model will generate images with homogeneous regions larger than the size of individual patches or regions that have smooth boundaries extending across multiple patches.

From a generative point of view this is certainly a drawback of the model. A hierarchical, recursive formulation of the field of masked RBMs provides an elegant solution to this problem as will be briefly discussed in section 5.4.2.

5.4 Discussion

.

In this chapter we have presented an extension of the masked RBM with occlusion shape model that is able to efficiently model larger images. The FoMRBM covers the full image with small partially overlapping latent patches that compete to explain the image pixels. Each individual latent patch has an associated shape and appearance and sets of overlapping patches have an associated relative depth ordering with overlapping shapes occluding each other. We have demonstrated that by keeping the size and the number of locally overlapping latent patches small, inference remains tractable even for larger images. We have evaluated the model on two datasets, a simple toy dataset and natural images. Experiments on the toy dataset demonstrate that learning and inference are feasible in principle. Experiments on natural images show that the model indeed is able to learn about shapes in more complex settings and that it can be applied to simple image processing tasks.

Figure 5.14: **Samples from the field of masked RBMs trained on natural images**. The sample images are of size 80×80 pixels and are obtained by sampling shape, appearance and depth independently for each of the 144 superpixels and then composing these patches according to their relative depth. The layout of the superpixels is as described in Fig 5.2. Individual samples from the shape model used to generate the above images are shown in Fig. 5.15.

5.4.1 Limitations of the FoMRBM

The FoMRBM in its current form is limited in several ways that we will discuss in detail below. Some these limitations have already been alluded to in the presentation above.

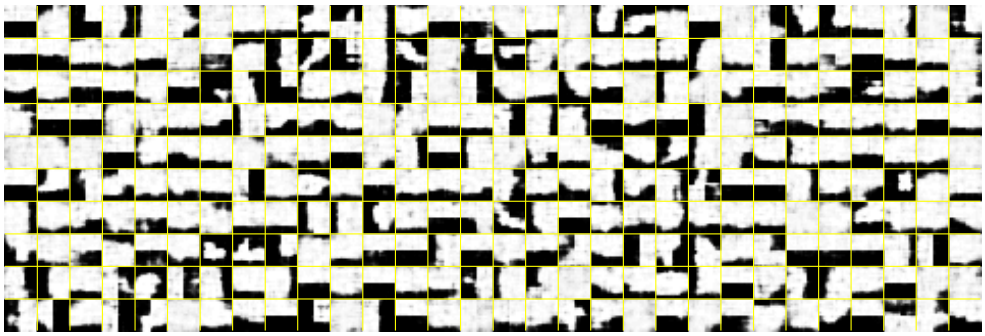


Figure 5.15: **Samples from the shape model trained on natural images**. The figure shows some of the shape samples (16×16 pixels) from the binary shape RBM that have been used to generated the full samples shown in Fig. 5.14.

5.4.1.1 Modeling Clutter and the "Background"

In the current formulation of the model, every image pixel has to be explained by one of the latent patches. This is not necessarily desirable. For instance, an image might be corrupted by noise, or, in certain applications one might only be interested in modeling certain aspects of an image (e.g. modeling a single foreground object) while other parts correspond to irrelevant clutter that is expensive to model but irrelevant for the task at hand. One solution to this problem would be to introduce an explicit "outlier" layer as discussed for the masked RBM in section 4.6.2.2 of the previous chapter (cf. equation (4.39)). The appearance model for such an outlier layer could be chosen to be a pixel-independent uniform distribution – or e.g. a color histogram suitable for the class of images to be modeled. Guo et al. (2003) (see discussion in section 5.2.4) use such an approach to deal with image pixels not covered by any texton.

Such a background/outlier layer could be placed either in front of the image or in the rear, with different implications for the shapes: If the outlier layer was assumed to be in front of all superpixels in the image, then any image pixel explained by that layer would render the corresponding shape pixels of the patches overlapping with that pixel unobserved. Assuming that the background-layer is in the rear, i.e. behind all superpixels, would require the shapes of all patches to be off whenever a pixel is assigned to that layer. Depending on the modeling goal, either approach (or both) could be appropriate.

Placing the outlier layer in the rear (i.e. behind all latent patches) would be suitable for modeling background clutter. In particular, it would solve the problem with the current formulation of the FoMRBM that was pointed out in section 5.1.2: Since a

pixel unexplained by any object could now be explained by the "background" layer, the rear-most latent patch would no longer be forced to be on if an image pixel is explained by none of the preceding patches. Assuming that the background layer is assigned mask-index $L+1$, and that $\pi(L+1) > \pi(l) \;\forall\; l \in \{1 \dots L\}$ (this simply says that the background-layer is always behind all latent patches) then equation (5.2) becomes

$$P(\mathbf{m}, \mathbf{s}_{1..L}, \mathbf{h}^{(s)}_{1..L}, \pi) \propto P(\pi) \left(\prod_i \left[\delta(s_{m_i, r_{m_i}(i)} = 1) \right]^{\delta(m_i \leqslant L)} \prod_{l \in o(i): \pi(l) < \pi(m_i)} \delta(s_{l, r_l(i)} = 0) \right) \times \left(\prod_l \text{SHAPE}(\mathbf{s}_l, \mathbf{h}^{(s)}_l) \right).$$

The corresponding conditional distribution over the image $\mathbf{v}$ and the latent appearances given the mask (compare with equation 5.1) is

$$p(\mathbf{v}, \widehat{\mathbf{v}}_{1\dots K}, \mathbf{h}^{(a)}_{1\dots K} | \mathbf{m}) = \prod_{l=1}^{L} \text{APP}(\widehat{\mathbf{v}}_l, \mathbf{h}^{(a)}_l) \prod_i \left[U(v_i) \right]^{\delta(m_i = L+1)} \left[\delta(\hat{v}_{m_i, r_{m_i}(i)} = v_i) \right]^{\delta(m_i < L+1)}, \tag{5.7}$$

where $U(x)$ is the outlier distribution, and $r_l(i)$ is the function that returns for image pixel i the corresponding pixel $r_l(i)$ within patch l if patch l overlaps with image pixel i (otherwise the function is not defined; cf. eq. (5.2) above). This is very similar to equation (4.39) for the MRBM and in Heess et al. (2011) we have implemented such a background layer for the model discussed in section 4.6.2.3 of the previous chapter. In this formulation of the FoMRBM the shapes of all patches are indeed marginally independent.

Assuming the outlier layer to be *in front* of all latent patches would allow dealing with unmodeled occluders. This is the approach we have taken for selecting K in the context of the MRBM with occlusion shape model (cf. section 4.6.2.2 in the previous chapter) and it would be similar to Williams and Titsias (2004) who use such an approach during greedy learning of a layered image model. It would be a conceptually small extension to generalize this approach to a range of different noise models and to allow, for instance, for additive Gaussian noise.

5.4.1.2 Translation invariance

In the latent patch layout described in section 5.1.1 latent patches are laid out in an overlapping manner on an 8×8 grid. This endows the FoMRBM with a form of "coarse" translation invariance. Unlike, however, for instance, in the convolutional

deep belief network (Lee et al., 2009), translations of "objects" by a small number of pixels are not being accounted for explicitly. This means that small shifts need to be learned by the shape model (and potentially also the appearance model) which is undesirable for at least two reasons: Firstly, having to learn about small shifts of the same shape is, in a sense, a waste of representational capacity of the shape RBM. Secondly, small translations of a shape can lead to vastly different representations in the hidden units which is undesirable in terms of the interpretability of this representation and makes it harder to extend the model hierarchically. Indeed, in preliminary experiments with toy data similar to the one used in section 5.3.1 but with small translations of the shapes within the receptive fields of the latent patches, the model's ability to recover the true shapes was noticeably impaired.

There are two conceivable solutions to this problem. One possibility would be to increase the number of latent patches and cover the image more densely. This would, however, lead to a larger number of overlapping latent patches making depth inference considerably more expensive. Furthermore, having a larger number of latent patches to explain a given image would likely worsen the over-segmentation problem encountered anyway (leading to less stable latent representations, see also discussion in section 5.4.1.3 below), and, more generally, would be wasteful (computationally and in terms of memory usage) because in many case a much smaller number of latent patches would be sufficient to explain an image.

A second, more promising possibility would be to take an approach similar to Lee et al. (2009) and to allow latent patches to be translated by a small number of pixels so that they can align with the local structure in an image. (This second approach is, in fact, similar to the first proposed solution if combined with a strong sparsity prior that limits the number of latent patches that can be active in the neighborhood of a pixel.) Inference in this model will require, for each latent patch, to explore not only different depths relative to its neighbors but also different positions. One downside of this approach is that inference of the positions cannot be conditioned on the mask. Instead, the (local) mask will have to be re-inferred when a latent patch is moved to a different position. This is potentially expensive, although keeping the state of neighboring superpixels fixed will hopefully lead to fast convergence of the local mask. We have started exploring this approach.

This second approach might also help solving another, related problem encountered in the current model: Different initializations of the mask can lead to rather different segmentation results (and thus inference in general). This is due to the blocked Gibbs

sampling scheme. For instance, in the case of the toy data from section 5.3.1 it might happen that the pixels belonging to an "object" (i.e. to one of the shapes) will not be associated with the latent patch that would be aligned with that object but with two of the neighboring patches that partially overlap with the object. Since latent appearances and latent shapes will be sampled conditioned on the mask, and the mask conditioned on the latent representation the model is unlikely to recover from this initial "error" when the Markov chain is run for number of steps that is realistic in practice and the object would be explained in terms of two halves represented in two latent patches. In our experiments we were able to largely circumvent this problem by initializing the mask to the center pixels of each latent patch, but this might not always be appropriate. Allowing for more flexibility when re-inferring the mask might alleviate this problem. (An alternative solution might be to use some annealing scheme when re-sampling the mask early during inference.)

5.4.1.3 Nature of the latent representation

One obvious limitation of the model has already been briefly discussed in section 5.3.2: The maximum distance over which correlations can be modeled is limited by the size of a latent patch. During inference, this means that larger regions are broken up into relatively small superpixels. Furthermore, the small size of the latent patches together with the fact that they overlap only partially means that relatively little evidence is available to infer the relative depth order of neighboring patches. When new images are sampled from the model it rarely generates homogeneous regions larger than the size of a superpixel, and for inpainting experiments the size of the latent patch is limiting the local evidence available to continue e.g. a region boundary.

As briefly discussed above, the problems during inference are compounded by the fact that frequently there are more latent patches available locally than would be strictly necessary to explain a certain part of an image (e.g. in largely homogeneous image regions) leading to an over-segmentation even relative to the size of the latent patches. Especially for complex data sets, such as natural images, where the distribution over plausible shapes is rather broad and the relative depth ordering of neighboring superpixels relatively uncertain, this will further increase the variability of the latent representation. This oversegmentation causes problems not only during inference, but especially also during learning: In our unsupervised learning scheme the inferred shapes are used to train the shape model in an EM-like manner. A largely arbitrary segmentation of homogeneous regions in the best case significantly slows down learning, and in

the worst case leads to a bias of the learned shape model: Although the model might initially only have a very small bias how to segment homogeneous regions, since it uses the inferred segmentations as training data, such a bias can become self-reinforcing. For instance, it could chose to only turn on the upper left quadrant of each patch to fill a homogeneous region. Due to the layout of latent patches, the region would still be fully covered. This presumably explains the bias observed in Fig. 5.15 at least partially.

Several improvements to the model are conceivable that would address these issues. The one that we expect to have most leverage is the recursive hierarchical formulation that will allow to model correlations between superpixels and that will be discussed in some detail in section 5.4.2 below. The over-segmentation issue could further be partially addressed by identifying a way to select the locally required number of latent patches (e.g. such as to reduce the number of latent patches in image regions that are largely homogeneous). This is effectively the same problem as the selection of K (i.e. of the number of layers) in the masked RBM and as discussed in section 4.6.2.2 of the previous chapter a hard problem. One appealing property of the solution proposed for the MRBM in section 4.6.2.2 is that it might be applicable to the FoMRBM as well: Starting with an initially small number of latent patches new latent patches could be instantiated depending on how many pixels in their receptive field are currently assigned to the outlier component. Overall, the situation is, however, more complicated than for the MRBM since for any given image pixel explained by the outlier component several alternative latent patches could be instantiated.

Finally, it might be beneficial to introduce the possibility of placing two neighboring patches at the "same depth". If two latent patches were placed at the same depth, there would be no occlusion so that the shapes of both latent patches would have to be aligned with the segmentation boundary, thus leading to a more constrained representation than is the case when one superpixel can occlude the other.

5.4.2 Future Work: The Deep Segmentation Network

The experiments in section 5.3.2 have highlighted the limitations of the model that arise from the limited size of the latent patches, which is considerably smaller than much of the relevant structure in natural images. In this section we will give a brief outlook of how the FoMRBM can be extended hierarchically to obtain a model which can account for large-scale structure and that explains a given natural scene in terms of a few high-level causes that decompose into parts and subparts. This hierarchical

formulation is ongoing work and its working title is "Deep Segmentation Network" (DSN). The basic idea underlying the DSN is to model correlations between superpixels in a dynamic, tree-structured hierarchy, in which the higher layers in the hierarchy model the correlations between superpixels of the subjacent layers. One appealing property of this formulation is that the FoMRBM presented in this chapter provides all the essential machinery needed to achieve this hierarchical formulation.

To illustrate this idea consider the toy data shown in Fig. 5.16 (*left* panel) which is very similar to the data used in section 5.3.1 but with the important difference that now certain combinations of "objects" occur considerably more frequently than would expected by chance if all objects were independent (e.g. two vertically aligned blue triangles with a black square in between). The corresponding latent representation inferred by a FoMRBM with suitable shape and appearance model is shown in the middle panel of Fig. 5.16. The format is the same as in Fig. 5.8: The latent patches are laid out on a grid, according to their position in the image. Each cell in the grid visualizes the state of a latent patch in the form of the reconstruction of the associated latent appearance and shape from the corresponding sets of binary hidden units. Although the first-level FoMRBM can model the individual "objects", it cannot account for their co-occurrences because the latent patches are assumed to be independent.

The approach taken in the DSN is to consider this latent representation as an "image" in which each first-level latent patch corresponds to a pixel, and to model this image with a second-level FoMRBM, which then accounts for the correlations between first-level latent patches. The *observed image* is a 80×80 pixel image with three continuous-valued channels per pixel (RGB values). The *latent representation* can be thought of as an image that has considerably fewer pixels, one for each latent patch, (here 12×12) but in which pixels have a much higher dimensionality, namely one binary channel for each of the hidden units of the shape and the appearance models (e.g. 384 + 128 for the model used for the experiments on natural images). This view allows to use the formulation of the FoMRBM to model correlations between superpixels in the same way as the first level of the FoMRBM models correlations between image pixels. For this purpose the second-layer image is covered by partially overlapping 4×4 patches in effectively the same way as the original (observed) image was covered by 16×16 patches (cf. Fig. 5.16 left and right panel). Each of these (second-level) latent patches will then be modeled by suitable shape and appearance models in the same way as the first-layer latent patches:

In the first level FoMRBM the appearance RBM is continuous valued and the shape

RBM binary. In the second level both shape and appearance models will be binary RBMs. As was the case for the first level the appearance RBM accounts for the state of the pixels in each latent patch. The latent patch is now of dimensionality 4×4 and each "pixel" in this patch has $N^{(1)}_{HS} + N^{(1)}_{HA}$ binary channels where $N^{(1)}_{HS}$ and $N^{(1)}_{HA}$ are the numbers of hidden units of the first layer shape and appearance models respectively (remember that each "pixel" in the second-level "image" corresponds to the concatenated set of hidden units – shape and appearance – of the first-level latent patch). Thus, the total number of visible units of the appearance model is $16 \times (N^{(1)}_{HS} + N^{(1)}_{HA})$. (For comparison, the first-level appearance model for RGB images with 16×16 patches has $256 \times 3 = 768$ continuous valued visible units.) The second-level shape model, too, plays the same role as for the first level and it accounts for the association of first-level superpixels with second-level latent patches, i.e. generatively it determines which of the 16 first-level superpixels generated by a second-level latent patch will actually be visible in the image. Since each second-level latent patch has only 16 "pixels", the shape RBM will be low-dimensional, e.g. 16 visible units for the 4×4 patches in the right-hand panel of Fig. 5.16. Second layer latent patches will group first layer latent patches in the same way as first layer latent patches grouped pixels in the original image into superpixels. This grouping of latent patches into higher-level superpixels will lead to a segmentation of the original image into larger, more coherent regions. This idea is illustrated for a simplified scenario in Fig. 5.17.

This formulation can be continued recursively until the full image is covered by a single set of aligned latent patches that determine the overall organization of the scene, giving rise to a tree structured hierarchy in which each lower-level (super) pixel is connected to exactly one higher-level pixel. Tree structured hierarchies have a long history in computer vision (e.g. Bouman and Shapiro, 1994; Luettgen and Willsky, 1995), although models with a fixed hierarchy are often not sufficiently flexible to align with the underlying structure in a given image. This has been addressed e.g. in the Dynamic Tree model (Williams and Adams, 1999; Storkey and Williams, 2003) which introduces additional flexibility: Instead of prescribing a particular tree it defines a prior over trees, so that the parse tree for any given image is determined by and can thus be aligned with the structure present in the image (for a similar idea see Hinton et al., 2000). The same will be true for the DSN: in each layer the mask determines which lower-level (super) pixel is associated with which higher-level superpixel, i.e. the mask determines the parent of the lower-level (super) pixel in the tree and thus plays the role of the parent indicator variable $\mathbf{z}$ in the Dynamic Tree model (Williams

and Adams, 1999). The DSN will, however, be able to learn significantly richer priors over tree structures than the Dynamic Tree model.

In the Dynamic Tree model a node in layer l does not have to be connected to a parent node in layer $l+1$ but can itself form the root of a tree. Thus, the Dynamic Tree model defines a prior over forests. Interestingly, introducing a background – or outlier – model as discussed in section 5.4.1.1 at every level of the hierarchy will similarly allow for an input dependent *forest* with roots of the trees at different levels of the hierarchy. This will effectively enable the DSN to model structured noise at different levels of complexity in a manner similar to ideas in compositional models (Jin and Geman, 2006; Bienenstock et al., 1996) in which parts can be bound into higher-level structures but can also remain unbound if such a binding would be unlikely.

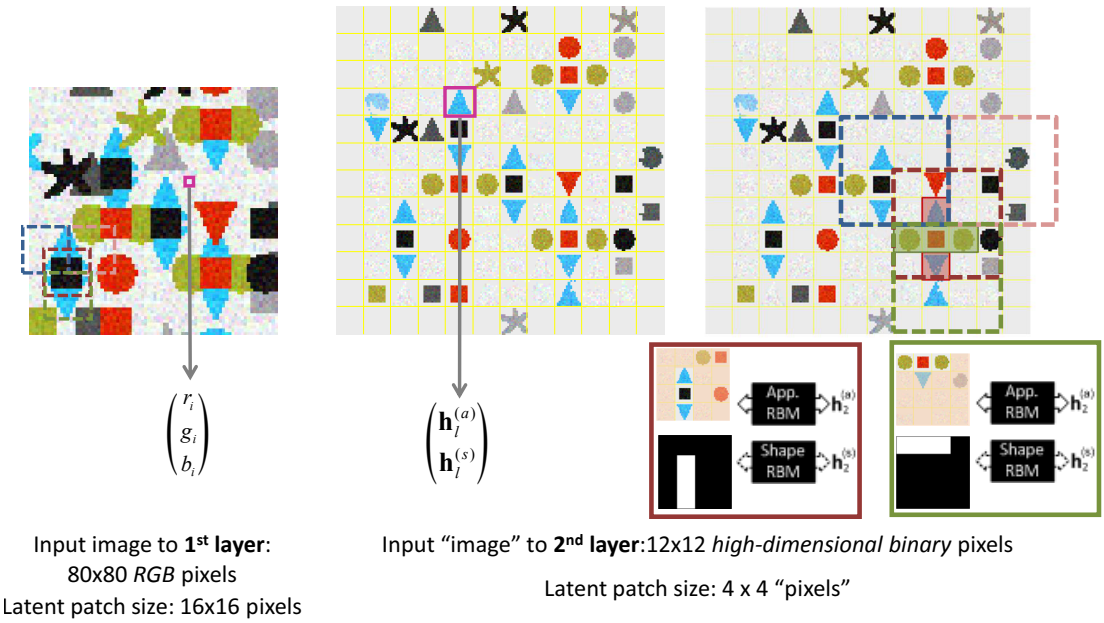


Figure 5.16: **Modeling superpixel correlations with a 2nd level FoMRBM**. *Left*: Toy data similar to the one used in section 5.3.1 (cf. Fig. 5.5). Each pixel in the image corresponds to a three dimensional vector for the red, green, and blue channels. The FoMRBM covers the image with overlapping 16×16 pixel latent patches (a subset of which is shown in the lower left corner: squares with dashed outlines). Latent patches model correlations between the image pixels within their receptive fields (for instance, the dark-red latent patch accounts for the correlations between the pixels representing the black square). Unlike for the data used in section 5.3.1, however, the objects in the image are no longer independent. Certain configurations of shapes occur more frequently than expected by chance (e.g. two vertically aligned blue triangles with a black square in between). These "superpixel-correlations" can be modeled by a *second layer FoMRBM*. *Middle*: The input to the second-layer FoMRBM is the latent representation inferred by the first layer. Latent patches are laid out according to their positions in the image. Each latent patch in the first level is considered as a "pixel" in a 12×12 image, and each such "pixel" corresponds to a high-dimensional binary vector that contains the state of the hidden units of the shape and appearance models of the respective patch. *Right*: The second-layer image is modeled by a FoMRBM. It is covered with overlapping 4×4 latent patches (a subset of which is shown, dashed squares). Each patch is modeled by a shape and an appearance model. The second-level appearance model models the state of the 16 first-level latent patches that make up a second level latent patch (their joint configuration of shape and appearance). The second-level shape model determines which first-layer superpixels are explained by a particular second layer latent patch. This is illustrated for two of the second level latent patches.

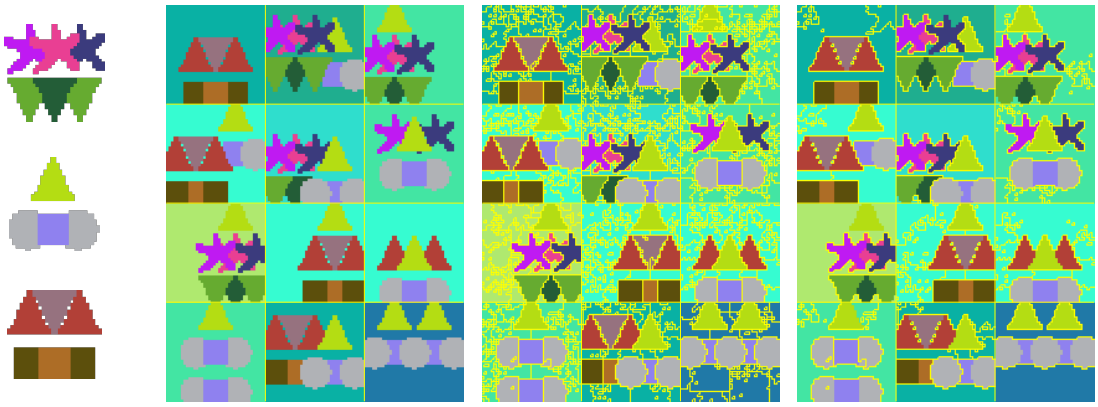


Figure 5.17: **Illustration of 2nd level segmentation**. To illustrate the grouping of 1st layer superpixels into more coherent 2nd layer superpixel we considered a simplified scenario. Small training images of size 48×48 pixels were generated to contain *groupings of shapes* used in the experiments described in section 5.3.1 *(a)*. Images were generated such that certain combinations of shapes occurred frequently *(b)*. For all training images we inferred a first layer latent representation and then trained a second-layer masked RBM with $K = 3$ (i.e. with 3 fully aligned latent patches) on the resulting set of latent images (for the results shown here we only trained a second-layer appearance model but no shape model). *(c)* shows the segmentation into superpixels obtained with the first-layer FoMRBM only. The results are very similar to the results in section 5.3.1 (cf. Fig. 5.12): individual shapes are separated out, and the background is over-segmented in a largely arbitrary manner. *(d)* shows the segmentation obtained after inference in the trained 2nd-layer masked RBM. Segmentation boundaries are now only shown where image pixels are associated with *different 2nd-layer latent patches*. While the model is clearly not entirely successful, the 2nd-layer model nevertheless tends to group large parts of the homogeneous background into a single superpixel. It has also to some extent learned to identify the three dominant combinations of shapes in the training images and in several cases groups them successfully during inference.

Chapter 6

Conclusion

6.1 Summary

The goal of this thesis has been to develop generative models of mid-level structure in natural images. The general modeling approach that we have adopted is to represent an image in terms of regions, and to explicitly model the regions' shapes and appearances.

6.1.1 Extended Fields of Experts

In chapter 3 we have focused on region appearance. We investigate the ability of the Field-of-Experts, a popular prior for generic image structure, to model image textures. We find that the basic formulation of the FoE is not able to model texture well, but that a more flexible, bimodal version of the expert function gives rise to a translation invariant, fully parametric, probabilistic generative model that is considerably more successful at this task. We evaluate this model on a texture synthesis and on an inpainting task. In a more detailed analysis of the different model formulations we try to provide some insight into where the differences in generative power arise: The bimodal expert function gives rise to a multi-modal distribution and one possible explanation is that different instantiations of a particular texture correspond to different modes of this distribution. Finally, we illustrate how this texture model could be used to obtain a more comprehensive model of multi-region images: In the region-based BiFoE we employ a mask which allows us to compose an image from multiple independent regions with different textures.

6.1.2 Masked RBM

In chapters 4 and 5 we have focused on region shape. In chapter 4 we extend work by collaborators, the masked RBM, with a model of region shape. We compare two alternative model formulations that can be used in this framework, the softmax model and the occlusion model. The occlusion shape model gives rise to a full model of image patches with a very intuitive generative process: An image patch is composed from multiple independent objects each with its own shape and appearance, and these objects are ordered with respect to their relative depth and interact in an occluding manner. This model explicitly accounts for the partial occlusion of objects and allows to reason about their relative depths. We propose efficient inference and unsupervised learning schemes and demonstrate that, when trained on natural image patches, the model captures important properties of this data well.

6.1.3 Field of masked RBMs

Explicit reasoning about occlusion and depth is expensive. One major limitation of the masked RBM with occlusion shape model developed in chapter 4 is that in its basic formulation it is limited to small image patches and a small number of occluding objects. In chapter 5, we therefore develop the Field of masked RBMs, which extends the basic masked RBM to larger images by composing them from many small objects which are partially overlapping and occluding. Many of the basic principles for inference and learning remain the same in this formulation, and the model can be trained in an unsupervised manner on large images. We demonstrate that when trained on natural images it learns about certain properties of region shape in images such as coherence and smoothness, and we further demonstrate its application to simple image processing tasks. One important limitation of the FoMRBM is that it decomposes the image into many small, independent objects and thus fails to capture longer-range structure in images. At the end of chapter 5 we therefore discuss a recursive, hierarchical extension of the model, the Deep Segmentation Network, in which the correlations between the small regions in the first level of the hierarchy are modeled in the higher levels. This recursive formulation arises from re-applying the basic formulation of the FoMRBM in each level in the hierarchy.

6.2 Discussion

6.2.1 Relationship between the models

In focusing on image regions the three pieces of work presented in this thesis, i.e. the BiFoE model (chapter 3; including the hierarchical, region-based BiFoE) and the Masked RBM / Field of Masked RBMs (chapters 4 and 5) share the overall modeling approach. The chapters are complementary in that chapter 3 primarily focuses on appearance, whereas chapters 4 and 5 focus on shape. There are technical similarities between the models in that their basic components are undirected graphical models, that the composition of different regions is in all cases achieved through a mask, and that region shape and region appearance are modeled independently (except for the model described in section 4.6.2.3).

The masked RBM is currently not translation invariant and it is effectively limited to small image patches and a small number of regions. The region-based BiFoE and the FoMRBM can both be applied to larger images and incorporate some form of stationarity although this is achieved in very different ways in the two models: The region-based BiFoE uses a homogeneous, i.e. fully translation invariant MRF for modeling region texture (and also for the naïve shape prior), and it allows for regions of arbitrary size. In contrast, in the FoMRBM we only have a coarse form of translation invariance (which is achieved by replicating latent patches in a regular manner across the image), and the size of regions is inherently limited by the size of the latent patches. As a consequence, in the FoMRBM a segmentation is in almost all cases an oversegmentation and the elementary regions (or "superpixels") need to be grouped into larger, more meaningful units using additional modeling levels as e.g. in the DSN. This oversegmentation is not desirable but the approach has certain advantages: In particular, it allows representation of an image in terms of a very large number of regions at a moderate computational cost. Secondly, spatially distant parts of an image tend to correspond to different visual entities so that limiting the extent over which correlations are modeled at the first level is not completely unreasonable, although the hard cut-off in the size of the superpixels and the fact they are arranged on a fixed grid is clearly undesirable. Finally, it is often not clear how different parts of an image should be grouped when only very local, low-level information is available. The DSN allows this grouping decision to be made in a flexible manner at different levels in the hierarchy. This allows the DSN to account for larger regions, and naturally gives rise to a segmentation hierarchy which, in many cases, is likely to be more appropriate than a

single segmentation.

6.2.2 Contour vs. region based representations

The region-based representations of images that we have considered in this thesis do not have an explicit notion of contour. Contours arise in the generated images at a region boundaries but are not directly modeled. This is different from much of the classical computer vision literature which relies on edges and their grouping into contours from which region boundaries can be deduced. While the formulation in terms of regions instead of contours seems plausible in light of the image formation process (many edges and especially extended contours in an image are indeed a consequence of occluding or abutting surfaces), it is not clear whether a region or a contour based representation would be more appropriate for many image interpretation problems. Further possibilities would be to combine contour and region-based representations (similar to Tu and Zhu, 2006; Guo et al., 2007), or, for instance, to employ a region-based representation for the generative component of a model while edge and contour extraction would be used for recognition and fast inference.

6.2.3 Distributed vs. structured representations

One might further ask whether the structured representation pursued in this thesis is generally a sensible way forward. It was originally motivated by the fact that very simple, generic image models have been struggling to capture important properties of images well. Very recently, however, several models have been proposed that address this problem and make progress towards generating richer generic image structure such as edges and region boundaries, without making this structure explicit in their formulations (e.g. Ranzato et al., 2010b, 2011; Courville et al., 2010). On the other hand, there are approaches attemping to represent image structure at a similar level as in this thesis and which employ even more highly structured representations (e.g. the work by Zhu and coworkers Guo et al., 2003; Zhu et al., 2005; Tu and Zhu, 2006; Guo et al., 2007, in which an image is composed from a large number of parameterized image primitives such as edge-elements or textons). This raises the question of how much structure to impose on the representation a priori. Several points should be noted in this context: Firstly, as discussed in section 4.4.1, it is interesting to observe that the mcRBM and related models (e.g. Ranzato et al., 2010b, 2011) implement principles similar to the ones sought after with the region-based representation, in particular, the

explicit modeling of the breakdown of correlations between image pixels across region boundaries and edges. Secondly, even though these novel models are considerably better at generating edge-like structure in images than, for instance, the FoE in its original formulation, they still fall short of modeling important properties of natural images such as the presence of textured regions. This suggests that these more generic models have not yet fully overcome the limitations of their predecessors. Thirdly, the main disadvantages of imposing too much structure on the representation is that this approach makes strong assumptions about the properties of natural images which are then engineered into the model formulation. This requires more effort from the designer of the model, the process is prone to mis-specification, and the resulting models are usually more complicated and less flexible. On the other hand, making certain properties of the structure in an image explicit should be seen as an advantage rather than a nuisance: the region-based models provide an explicit representation of the decomposition of the image into regions; the MRBM and FoMRBM further allow to reason explicitly about image depth and occlusion. This information is not directly available from the representation of the mcRBM and related models and would have to be extracted from their latent representation by other means (if possible). One interesting property of the models presented in this thesis is that they combine a generic, distributed representation with a notion of identifiable visual entities – the regions or superpixels. The DSN further combines a distributed representation with a hierarchy that enforces a single parent constraint for *groups* of hidden units and makes explicit the fact that images can be composed from independent visual entities (e.g. the different objects in a scene).

6.2.4 Generative models and unsupervised learning

In this thesis we have relied on generative models for capturing some of the structure in natural images. The main advantages of this approach are the generality of the models that are obtained (the masked RBM, in a single model, allows to segment image patches, reason about the latent shapes and depth of the constituting objects, and to generate new images), their ability to explicitly handle uncertainty (in the context of the masked RBM, for instance, with respect to the depth ordering), and the suitability for unsupervised learning. Also, depending on the model formulation, this approach forces us to make explicit the assumptions that underlie the model (a good example of this is the composition process in the MRBM / FoMRBM, although in other models the assumptions are somewhat less clear e.g. in the BiFoE) and procedures for inference

and learning directly follow from the definition of the generative model. One major downside of this approach is that exact inference can be very expensive, as we have indeed experienced in this thesis (especially in chapters 4 and 5). Alternative learning frameworks such as energy-based models (e.g. LeCun et al., 2006) that avoid some of the complexities of a fully probabilistic interpretation (especially the need to normalize) but can still capture complicated dependencies in the data have been applied to other learning proplems in vision. They have been used, for instance, to learn, in an unsupervised manner, feature representations for object recogntion (e.g. Ranzato et al., 2007). While they might be applicable to formulations similar to the models presented in this thesis they would lack at least some of the appealing properties of generative models, such as their ability to handle uncertainty and to generate new samples.

We have further focused almost entirely on unsupervised learning from static images. Unsupervised learning is conceptually appealing and avoids the need to obtain expensive labeled data. On the other hand, in complex models it can be difficult and carefully chosen learning strategies are often required (cf. chapters 4 and 5, especially section 4.6.2.3). Generative models are, however, not limited to unsupervised learning and semi-supervised strategies (e.g. including a few segmented images in the case of the masked RBM) might help to make learning more efficient and / or robust. Also, the unsupervised learning problem can potentially be made easier if richer (but still unlabeled) training data is used: in the case of the masked RBM, for instance, spatio-temporal data might allow exploitation of motion cues to identify independently moving objects which would make additional information regarding segmentation and relative depth available to the learning algorithm.

6.2.5 Connection to biological vision

Unlike some other work in the deep learning community this thesis is entirely focused on the problem of modeling natural image structure and does not attempt to provide an explanation of the computational principles epmployed by the human visual system. In fact, the models proposed in this thesis lack features such as units resembling simple or complex cells that are commonly seen as important properties of biologically more plausible architectures. Nevertheless the MRBM and the FoMRBM have some of the ability of the human visual system to reason about occlusion and partially occluded shapes (obviously in a much simpler manner). This gives rise to simple perceptual illusions such as the Kanisza triangle, although the underlying computational principles

are likely to be quite different. The separation of an image into depth layers in the MRBM / FoMRBM also bears some – albeit remote – similarity to the surface-based representation proposed in the cognitive science literature (Nakayama et al., 1995).

6.3 Future Work

We have already discussed various directions for future work in the Discussion sections of chapters 3, 4, and 5. In section 6.3.1 we will therefore only briefly summarize the most important points and then take a somewhat broader perspective in section 6.3.2.

6.3.1 Extensions of the models discussed in this thesis

One of the major limitations of the BiFoE texture model developed in chapter 3 is the fact that it currently requires a separate model to be trained for each texture and an important future extension would be a latent variable formulation in which different textures are specified by different configurations of the latent variables and that allows learning a model of a large number of different textures with a *single* set of parameters. Furthermore, in order to turn the region-based BiFoE into a valid model of generic image structure, richer priors over region shape and an efficient way to deal with a large number of regions will be required.

The most obvious direction for future work in the MRBM / FoMRBM framework is the implementation of the Deep Segmentation Network that we have outlined at the end of chapter 5. This formulation, we hope, will give rise to a compositional, hierarchical representation of natural image structure that accounts for an important property of the image formation process, occlusions, and for the fact that images are typically composed from several independent entities. In addition to this long-term goal, there are several technical challenges associated with the MRBM / FoMRBM framework. These include the problem of selecting the number of latent patches (i.e. selecting K; see also section 4.6.2.2) but also the development of more efficient inference schemes. Regarding the latter, one promising direction would be the use of discriminative methods to speed up inference in our generative model as e.g. in Dayan et al. (1995) or Tu et al. (2001). A further potential direction for future work would be to incorporate additional invariances (especially with respect to translation) into the FoMRBM.

Finally, all models considered in this thesis contain undirected components. Despite considerable attention in the literature (cf. discussion in chapter 2.3.2) learning

undirected models remains problematic and requires many hyper-parameters to be chosen manually. Developing faster and more robust methods will therefore be important for the future use of these models, especially in large scale applications.

6.3.2 Towards richer models of image structure

Taking a somewhat more general perspective and focusing on the goal of developing more comprehensive models of natural image structure, there are several interesting directions for longer-term research. One especially exciting direction would be the development of model formulations that can capture more of the physical and semantic properties of a visual scene. One aspect of this problem is the ability to reason about the shape, orientation, illumination, and material properties of occluding surfaces, similar e.g. to the 2.5D sketch model proposed by Marr (1977). Other aspects are, for instance, the ability to capture high-level structure such as information about parts, objects, and object categories, the ability to reason about the 3D geometry of a scene and of the objects contained in it, and also the problem of dealing with invariances e.g. with respect to location, or scale. All these problems have been considered in some (or many) forms in the computer vision literature, although not necessarily in a manner satisfying with respect to the goal of learning a comprehensive model of image structure. The first aspect seems especially interesting in that the ability to reason about surfaces would seem like a natural extension of the work in this thesis. Also, although related problems have received considerable attention in specialist work in the computer vision literature (for instance, in the shape-from-X literature; e.g. Zhang et al., 1999), there is relatively little work in the context of more general generative models of image structure (but see, e.g. Zhu et al., 2005; Han and Zhu, 2007).

These problems raise interesting questions related to the issues already briefly discussed in section 6.2 above, concerning the type of model architecture that is chosen to address these problems, and the learnability. For instance, in chapters 4 and 5 we have developed a model that explicitly reasons about occlusion and relative image depth. Will a similarly explicit model formulation be necessary to successfully reason and learn about other aspects of the physical environment, or could a sufficiently rich generic learning architecture capture many of these properties of a scene implicitly, and make them accessible in a suitable manner without the need for specialist formulations? Similarly, the predominant approach in the literature to modeling higher-level structure are parse-trees of some form, that capture explicitly the grouping of parts

into larger entities (e.g. Zhu and Mumford, 2006). This reflects our intuitive understanding of the world and makes the latent representation inferred for an image easily interpretable. On the other hand it presupposes much of how visual structure should be represented. It is not clear whether this is necessary or whether at least some of this representational structure can be discovered (i.e. learned from data; this problem has received some attention in the cognitive science literature, e.g. Tenenbaum et al., 2011 and references therein). For instance, it is conceivable that tree-like part-whole relationships could be learned and embedded in a general-purpose distributed representation. An interesting, related example is that of transformations: Transformations, such as translations or scale are hard-wired into the formulation of many models (including some models in the deep learning literature, e.g. Lee et al., 2009). Recent work shows, however, that such transformations can be learned in an unsupervised manner using a generic architecture if suitable training data is provided (e.g. Memisevic and Hinton, 2007).

If one believes that it might be possible to capture many important properties of natural images in a general purpose architecture, then one might wonder what this architecture should look like. Deep learning architectures such as RBMs, DBNs, and DBMs would seem to be natural candidates, but the work in this thesis and elsewhere suggests that, at least in their naïve formulation, they might still be rather inefficient at modeling certain types of structure. One potentially more powerful learning component are three-way (or higher-order) Boltzmann machines which can be thought of as RBMs in which the connections between visibles and hiddens are gated by a third set of latent variables and which can be thought of as an efficient mixture of a large number of RBMs with different parameters. They have recently been applied to a range of interesting problems (including the modeling of transformations in, Memisevic and Hinton, 2007, or to recognition under occlusion, Tang, 2010) and also underly the mcRBM discussed above.

As the nature of the represented structure moves away from the visual properties of an image and towards the physical or semantic properties of the underlying scene, and as the model formulation becomes more complicated, learning is likely to be greatly aided by going beyond unsupervised learning from static images. As discussed above, this could include the use of partially labeled data (e.g. semi-supervised approaches). More appealing would, however, be the use of richer data that is naturally available to an agent interacting with the visibile world such as spatio-temporal or stereo data, together with suitable learning criteria. Such data can provide information regarding

the decomposition of a scene into independent objects, and spatio-temporal data can further help, for instance, with the learning of transformations and invariant representations (Memisevic and Hinton, 2007 learn transformations from pairs of related images where one is a transformed version of the other; for invariances see e.g. Becker, 1996; Mobahi et al., 2009). Overall, and in line with the discussion in previous paragraphs, it seems important to focus on models and suitable learning techniques that allow for the acquired "knowledge" to be re-applied in novel situations and to novel tasks, an idea that is also referred to as "transfer learning".

Bibliography

K. Abend, T. J. Harley, and L. N. Kanal. Classification of binary random patterns. *IEEE Transactions on Information Theory*, 11(4):538–544, 1965.

D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147 – 169, 1985.

N. J. Adams and C. K. I. Williams. Dynamic trees for image modelling. *Image and Vision Computing*, 21(10):865 – 877, 2003.

E. H. Adelson. Layered representations for image coding. Technical report, Media Laboratory, MIT, 1991.

H. Barlow. Unsupervised learning. *Neural Computation*, 1(3):295–311, 1989.

S. Becker. Learning temporally persistent hierarchical representations. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 824–830, 1996.

A. Bell and T. Sejnowski. The "Independent Components" of Natural Scenes are Edge Filters. *Vision Research*, 37:3327–3338(12), 1997.

Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

Y. Bengio and O. Delalleau. Justifying and generalizing contrastive divergence. *Neural Comput.*, 21(6):1601–1621, 2009.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. ACM.

P. Berkes, R. Turner, and M. Sahani. On sparsity and overcompleteness in image models. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2007.

M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

M. Bethge. Factorial coding of natural images: how effective are linear models in removing higher-order dependencies? *J. Opt. Soc. Am. A*, 23(6):1253–1268, 2006.

E. Bienenstock, S. Geman, and D. Potter. Compositionality, MDL priors, and object recognition. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 838–844, 1996.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.

L. Bottou. Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004. URL `http://leon.bottou.org/papers/bottou-mlss-2004`.

G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 710–715, Washington, DC, USA, 2005. IEEE Computer Society.

C. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162 –177, March 1994.

P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.

R. Buccigrossi and E. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8(12):1688 –1701, December 1999.

N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. Interpreting image databases by region classification. *Pattern Recognition (Special Edition on Image Databases)*, 30(4):555–563, April 1997.

M. A. Carreira-Perpiñ and G. Hinton. On contrastive divergence learning. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005*, pages 33–40. Society for Artificial Intelligence and Statistics, 2005.

R. Chellappa, S. Chatterjee, and R. Bagdazian. Texture synthesis and compression using Gaussian-Markov random field models. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:298–303, 1985.

A. Courville, J. Bergstra, and Y. Bengio. Modeling natural image covariance with a spike and slab restricted Boltzmann machine. In *NIPS Workshop on Deep Learning*, 2010.

T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, September 2004.

G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1):25–39, January 1983.

G. E. Dahl, M. Ranzato, A. Mohamed, and G. E. Hinton. Phone recognition with the mean-covariance restricted Boltzmann machine. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 469–477. 2010.

P. Dayan and R. S. Zemel. Competition and multiple cause models. *Neural Computation*, 7(3):565–579, 1995.

P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural Computation*, 7:889–904, September 1995.

J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 361–368, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55, 1987.

G. Desjardins and Y. Bengio. Empirical evaluation of convolutional RBMs for vision. Technical Report 1327, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 2008.

G. Desjardins, A. Courville, and Y. Bengio. Parallel tempering for training of restricted Boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 145–152, 2010.

J. Domke, A. Karapurkar, and Y. Aloimonos. Who killed the directed model? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, June 2008.

J. Dorsey, A. Edelman, H. W. Jensen, J. Legakis, and H. K. Pedersen. Modeling and rendering of weathered stone. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 225–234, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195:216–222, 1987.

A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR 1999)*, pages 1033–1038, Corfu, Greece, September 1999.

B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Washington, DC, USA, 2007. IEEE Computer Society.

P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal Of Computer Vision*, 61(1):55–79, 2005.

S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Washington, DC, USA, 2007. IEEE Computer Society.

A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV '03)*, Washington, DC, USA, 2003. IEEE Computer Society.

W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

Y. Freund and D. Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. Technical Report UCSC-CRL-94-25, University of California, Santa Cruz, 1994.

B. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1392 –1416, September 2005.

B. J. Frey and N. Jojic. Learning appearance and transparency manifolds of occluding objects in layers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)*. IEEE Computer Society Press, Los Alamitos, CA, 2003.

N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 125–133, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

M. Frydenberg. The chain graph Markov property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.

A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3): 367–383, 1992.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721 –741, November 1984.

S. Geman and D. E. McClure. Bayesian image analysis: An application to single photon emission tomography. In *Proceedings of the Statistical Computing Section, American Statistical Association (1985)*, pages 12–18, 1985.

C. J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Proceedings of the 23rd Symposium on the Interface: Computing Science and Statistics*, New York, 1991. American Statistical Association.

C. J. Geyer and E. A. Thompson. Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3):657–699, 1992.

Z. Ghahramani. Factorial learning and the em algorithm. In *Advances in Neural Information Processing Systems 7*, pages 617–624. MIT Press, 1995.

G. L. Gimel'farb. Texture modeling by multiple pairwise pixel interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1110–1114, 1996.

M. Girolami. A variational method for learning sparse and overcomplete representations. *Neural Computation*, 13(11):2517–2532, 2001.

I. Goodfellow, Q. Le, A. Saxe, H. Lee, and A. Ng. Measuring invariances in deep networks. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 646–654. 2009.

A. Gray, J. Kay, and D. Titterington. An empirical study of the simulation of various models used for images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):507 –513, May 1994.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

U. Grenander and A. Srivastava. Probability models for clutter in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):424 –429, April 2001.

C.-E. Guo, S.-C. Zhu, and Y. N. Wu. Modeling visual patterns by integrating descriptive and generative methods. *International Journal of Computer Vision*, 53:5–29, 2003.

C.-e. Guo, S.-C. Zhu, and Y. N. Wu. Primal sketch: Integrating structure and texture. *Comput. Vis. Image Underst.*, 106:5–19, April 2007.

M. Gutmann and A. Hyvrinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, 2010.

F. Hamze and N. de Freitas. From fields to trees. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence (UAI 2004)*, pages 243–250, Arlington, Virginia, United States, 2004. AUAI Press.

F. Han and S.-C. Zhu. A two-level generative model for cloth representation and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (7):1230–1243, July 2007.

M. Hassner and J. Sklansky. The use of Markov random fields as models of texture. *Computer Graphics and Image Processing*, 12(4):357 – 370, 1980.

X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *Proceedings of the 9th European Conference on Computer Vision (ECCV 2006)*, pages 338–351. Springer, 2006.

D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pages 229–238, New York, NY, USA, 1995. ACM.

N. Heess, N. Le Roux, and J. Winn. Weakly supervised learning of foreground-background segmentation using masked RBMs. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Proceedings of the 21st International Conference on*

Artificial Neural Networks (ICANN 2011), volume 6792 of *Lecture Notes in Computer Science*, pages 9–16. Springer Berlin / Heidelberg, 2011.

G. Hinton. A practical guide to training restricted Boltzmann machines. Technical Report UTML TR 2010003, University of Toronto, 2010.

G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

G. Hinton, P. Dayan, B. Frey, and R. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

G. Hinton, B. Sallans, and Z. Ghahramani. A hierarchical community of experts. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic Publishers, 1998.

G. Hinton, S. Osindero, M. Welling, and Y. Teh. Unsupervised discovery of non-linear structure using contrastive backpropagation. *Cognitive Science*, 30(4):725–731, 2006a.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002.

G. E. Hinton and Y. W. Teh. Discovering multiple constraints that are frequently approximately satisfied. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 227–234, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

G. E. Hinton, Z. Ghahramani, and Y. W. Teh. Learning to parse images. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 463–469. MIT Press, Cambridge, MA, 2000.

G. E. Hinton, M. Welling, and A. Mnih. Wormholes improve contrastive divergence. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2003. MIT Press.

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006b.

D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24:577–584, July 2005.

B. K. P. Horn. Understanding image intensities. *Artificial Intelligence*, 8(2):201 – 231, 1977.

G. Huang, M. Rames, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report TR 07-49, Univ. of Mass., Amherst, 2007.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Hyvärinen. Some extensions of score matching. *Comput. Stat. Data Anal.*, 51: 2499–2512, February 2007.

A. Hyvärinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

A. Hyvärinen, P. O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

S. Jain and R. Neal. Splitting and merging components of a nonconjugate dirichlet process mixture model. *Bayesian Analysis*, 2:445–472, 2007.

D. Jeulin. Dead leaves models: From space tessellation to random functions. In D. Jeulin, editor, *Proceedings of the Interational Symposium on Advances in Theory and Applications of Random Sets, 1996*, pages 137–156. World Scientific Publishing Company, 1997.

Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, pages 2145 – 2152, 2006.

N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages 199–206. IEEE Computer Society Press, 2001. Kauai, Hawaii.

B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, 8 (2):84–92, february 1962.

A. Kannan, N. Jojic, and B. J. Frey. Generative model for layers of appearance and deformation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 166–173. Society for Artificial Intelligence and Statistics, 2005.

A. Kannan, J. Winn, and C. Rother. Clustering appearance and shape by learning jigsaws. In *In Advances in Neural Information Processing Systems*. MIT Press, 2007.

Y. Karklin and M. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457:83–86, January 2009.

Y. Karklin and M. S. Lewicki. Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14:483–499, 2003.

K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun. Learning convolutional feature hierarchies for visual recognition. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1090–1098. 2010.

M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1189–1197. 2010.

V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.

S. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are quantitative and some qualitative. *Annals of Statistics*, 17:31–57, 1989.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988.

N. Le Roux and Y. Bengio. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, 20(6):1631–1649, 2008.

N. Le Roux, N. Heess, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 23(3):593–650, 2011.

Y. LeCun. Generalization and network design strategies. In R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, editors, *Connectionism in Perspective*, Zurich, Switzerland, 1989a. Elsevier.

Y. LeCun. Generalization and network design strategies. Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, 1989b.

Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. In G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, editors, *Predicting Structured Data*. MIT Press, 2006.

A. Lee, D. Mumford, and J. Huang. Occlusion models for natural images: A statistical study of a scale-invariant Dead Leaves model. *International Journal of Computer Vision*, 41(1/2):35–59, 2001.

D. D. Lee and S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th International Conference on Machine Learning*, pages 609–616, 2009.

M. Lewicki and B. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A*, 16:1587–1601, 1999.

B. Lindsay. Composite likelihood methods. *Contemporary Mathematics*, 80:221–239, 1988.

J. Lücke and M. Sahani. Maximal causes for non-linear component extraction. *Journal of Machine Learning Research*, 9:1227–1267, 2008.

J. Lücke, R. Turner, M. Sahani, and M. Henniges. Occlusive components analysis. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1069–1077, 2009.

M. Luettgen and A. Willsky. Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination. *IEEE Transactions on Image Processing*, 4(2):194–207, February 1995.

B. Manjunath, T. Simchony, and R. Chellappa. Stochastic and deterministic networks for texture segmentation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(6):1039–1049, June 1990.

E. Marinari and G. Parisi. Simulated tempering: A new Monte Carlo scheme. *EPL (Europhysics Letters)*, 19(6):451, 1992.

B. Marlin, K. Swersky, B. Chen, and N. de Freitas. Inductive principles for restricted Boltzmann machine learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

D. Marr. *Representing Visual Information*, pages 61–80. Academic Press, 1977.

J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. Technical Report AIM-897, Massachusetts Institute of Technology, Cambridge, MA, 1987.

J. Martens and I. Sutskever. Parallelizable sampling of markov random fields. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 517–524, 2010.

J. McAuley, T. Caetano, A. Smola, and M. Franz. Learning high-order MRF priors of color images. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pages 617–624, New York, NY, USA, 2006. ACM.

D. Melas and S. Wilson. Double Markov random fields and Bayesian image segmentation. *IEEE Transactions on Signal Processing*, 50(2):357–365, February 2002.

R. Memisevic and G. Hinton. Unsupervised learning of image transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, 2007.

H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pages 737–744, New York, NY, USA, 2009. ACM.

G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR 2004)*, volume 2, pages 326–333, 2004.

R. Morris, X. Descombes, and J. Zerubia. The Ising/Potts model is not well suited to segmentation tasks. In *Proceedings IEEE Digital Signal Processing Workshop*, pages 263–266, September 1996.

I. Murray and R. Salakhutdinov. Evaluating probabilities under high-dimensional latent variable models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, 2009.

V. Nair and G. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, 2010.

K. Nakayama, Z. J. He, and S. Shimojo. *Visual Cognition*, chapter Visual surface representation: a critical link between lower-level and higher-level vision, pages 1–70. MIT Press, Cambridge, MA, 1995.

R. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

R. M. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353–366, 1996.

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman & Hall / CRC Press, 2011.

R. M. Neal and G. E. Hinton. *A view of the EM algorithm that justifies incremental, sparse, and other variants*, pages 355–368. MIT Press, Cambridge, MA, USA, 1998.

M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2735 –2742, June 2009.

B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325(15), 1997.

B. A. Olshausen and K. J. Millman. Learning sparse codes with a mixture-of-Gaussians prior. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 841–847. The MIT Press, 2000.

B. Ommer and J. M. Buhmann. Learning the compositional nature of visual object categories for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:501–516, 2010.

S. Osindero and G. Hinton. Modeling image patches with a directed hierarchy of Markov random field. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

S. Osindero, M. Welling, and G. Hinton. Topographic product models applied to natural scene statistics. *Neural Computation*, 18(7):344–381, 2006.

R. Paget and I. Longstaff. Texture synthesis via a noncausal nonparametric multiscale Markov random field. *IEEE Transactions on Image Processing*, 7(6):925 –931, June 1998.

S. E. Palmer. *Vision Science*. MIT Press, Cambridge, MA, 1 edition, 1999.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

M. Pihlaja, M. Gutmann, and A. Hyvärinen. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*, 2010.

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

K. Popat and R. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Visual Communications and Image Processing*, pages 756–768, 1993.

J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1): 49–70, 2000.

J. Puertas, J. Bornschein, and J. Lucke. The maximal causes of natural scenes are edge filters. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1939–1947. 2010.

R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine learning*, pages 759–766, 2007.

M. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 2551 –2558, June 2010.

M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*. IEEE Press, 2007.

M. Ranzato, A. Krizhevsky, and G. Hinton. Factored 3-way restricted Boltzmann machines for modeling natural images. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 621–628, 2010a.

M. Ranzato, V. Mnih, and G. E. Hinton. How to generate realistic images using gated MRFs. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, 2010b.

M. Ranzato, J. Susskind, V. Mnih, and G. Hinton. On deep generative models with applications to recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, 2011.

X. Ren. *Probabilistic Models for Mid-Level Vision*. PhD thesis, U.C. Berkeley, 2006.

X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (CVPR 2003)*, volume 1, pages 10–17, October 2003.

D. A. Ross and R. S. Zemel. Learning parts-based representations of data. *Journal of Machine Learning Research*, 7:2369–2397, December 2006.

S. Roth. *High-order Markov random fields for low-level vision*. PhD thesis, Brown University, 2007.

S. Roth and M. Black. Fields of Experts: a framework for learning image priors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2:860–867, June 2005.

S. Roth and M. Black. Fields of experts. *International Journal of Computer Vision*, 82:205–229, 2009.

C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1382–1389, June 2009.

D. L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37(23):3385 – 3398, 1997.

R. Salakhutdinov. Learning in deep Boltzmann machines using adaptive MCMC. In *Proceedings of the 27th International Conference on Machine Learning*, 2010a.

R. Salakhutdinov. Learning in Markov random fields using tempered transitions. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, 2010b.

R. Salakhutdinov and G. Hinton. Deep Boltzmann Machines. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.

R. Salakhutdinov and H. Larochelle. Efficient learning of deep Boltzmann machines. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 693–700, 2010.

R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine Learning*, volume 25, 2008.

L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. In *Advances in Neural Information Processing Systems 8*, pages 486–492. The MIT Press, 1996.

E. Saund. A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7(1):51–71, 1995.

U. Schmidt, Q. Gao, and S. Roth. A generative perspective on MRFs in low-level vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.

M. Seeger. Bayesian inference and optimal design in the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

F. Sinz and M. Bethge. The conjoint effect of divisive normalization and orientation selectivity on redundancy reduction. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1521–1528, Curran, Red Hook, NY, USA, 2009.

F. Sinz, E. P. Simoncelli, and M. Bethge. Hierarchical modeling of local image features through Lp-nested symmetric distributions. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1696–1704, Curran, Red Hook, NY, USA, 2010.

E. Smith and M. S. Lewicki. Efficient coding of time-relative structure using spikes. *Neural Computation*, 17(1):19–45, 2005.

P. Smolensky. *Information processing in dynamical systems: foundations of harmony theory*, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

A. J. Storkey and C. K. I. Williams. Image modeling with position-encoding dynamic trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):859–871, 2003.

I. Sutskever and T. Tieleman. On the convergence properties of contrastive divergence. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5:271–301, 1990.

Y. Tang. Gated Boltzmann machine for recognition under occlusion. In *NIPS Workshop on Transfer Learning by Learning Rich Generative Models*, 2010.

M. Tappen. Utilizing variational optimization to learn Markov random fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8, June 2007.

Y. Teh, M. Welling, S. Osindero, and G. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, 2003.

Y. W. Teh. *Bethe Free Energy and Contrastive Divergence Approximations for Undirected Graphical Models*. PhD thesis, Department of Computer Science, University of Toronto, 2003.

J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011.

L. Theis, S. Gerwinn, F. H. Sinz, and M. Bethge. In all likelihood, deep belief is not enough. *CoRR*, abs/1011.6086, 2010.

T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1064–1071, New York, NY, USA, 2008. ACM.

T. Tieleman and G. Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1033–1040. ACM New York, NY, USA, 2009.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

M. K. Titsias and C. K. I. Williams. Fast unsupervised greedy learning of multiple objects and parts from video. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 12*, page 179, Washington, DC, USA, 2004. IEEE Computer Society.

H. Tjelmeland and J. Besag. Markov random fields with higher-order interactions. *Scandinavian Journal of Statistics*, 25(3):415–433, 1998.

S. Todorovic and N. Ahuja. Unsupervised category modeling, recognition, and segmentation in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2158–2174, 2008.

Z. Tu and S.-C. Zhu. Parsing images into regions, curves, and curve groups. *International Journal of Computer Vision*, 69:223–249, August 2006.

Z. Tu, S.-C. Zhu, and H.-Y. Shum. Image segmentation by data driven Markov chain Monte Carlo. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (CVPR 2001)*, volume 2, pages 131–138, 2001.

D. Vickrey, C. Lin, and D. Koller. Non-local contrastive objectives. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1:1–305, January 2008.

J. Wang and E. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625 –638, September 1994.

G. C. G. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

Y. Weiss and W. Freeman. What makes a good model of natural images? *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8, June 2007.

M. Welling, G. Hinton, and S. Osindero. Learning sparse topographic representations with products of Student-t distributions. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1359–1366. MIT Press, Cambridge, MA, 2003.

M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems 17*, Cambridge, MA, 2004. MIT Press.

C. Williams and N. Adams. DTs: Dynamic Trees. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 634–640. MIT Press, 1999.

C. K. I. Williams and F. V. Agakov. Products of Gaussians and probabilistic minor component analysis. *Neural Computation*, 14(5):1169–1182, 2002.

C. K. I. Williams and M. K. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062, 2004.

J. M. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 756–763. IEEE Computer Society, 2005.

A. P. Witkin and M. Kass. Reaction-diffusion textures. *ACM Siggraph Computer Graphics*, 25:299–308, 1991.

L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In H. Zha, R.-i. Taniguchi, and S. Maybank, editors, *Computer Vision - ACCV 2009*, volume 5995 of *Lecture Notes in Computer Science*, pages 88–97. Springer Berlin / Heidelberg, 2010.

O. Woodford, I. Reid, P. Torr, and A. Fitzgibbon. Fields of experts for image-based rendering. In *Proceedings of the 17th British Conference on Machine Vision (BMVC 2006)*, 2006.

J. Woods. Two-dimensional discrete Markovian fields. *IEEE Transactions on Information Theory*, 18(2):232 – 240, March 1972.

S. Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pages 291–294, New York, NY, USA, 1996. ACM.

L. Younes. Parametric inference for imperfectly observed Gibbsian fields. *Probability Theory and Related Fields*, 82:625–645, 1989.

A. L. Yuille. The convergence of contrastive divergences. In *Advances in Neural Information Processing Systems 17*, 2004.

A. Zalesny and L. J. V. Gool. A compact model for viewpoint dependent texture synthesis. In *SMILE '00: Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 124–143, London, UK, 2001. Springer-Verlag.

M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus. Deconvolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 2528 –2535, June 2010.

J. Zhang, J. Modestino, and D. Langan. Maximum-likelihood parameter estimation for unsupervised stochastic model-based image segmentation. *IEEE Transactions on Image Processing*, 3(4):404–420, July 1994.

R. Zhang, P.-S. Tsai, J. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999.

L. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *Proceedings of the 10th European Conference on Computer Vision (ECCV 2008)*, pages 759–773, Berlin, Heidelberg, 2008. Springer-Verlag.

S. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259–362, 2006.

S. C. Zhu and D. Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11), November 1997.

S. C. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *International Journal of Computer Vision*, 62:121–143, April 2005.

Appendix A

Bimodal Field-of-Experts

A.1 Unimodality of Student-t FoE

Consider the energy of the Student-t *PoE*:

$$E(\mathbf{x}) = \sum_j v_j \log \underbrace{\left\{1 + \frac{1}{2}(\mathbf{w}_j^T \mathbf{x})^2\right\}}_{z_j(\mathbf{x})} \tag{A.1}$$

$$\frac{\partial E}{\partial x_i} = \sum_j \frac{v_j w_{ji}(\mathbf{w}_j^T \mathbf{x})}{z_j(\mathbf{x})} \tag{A.2}$$

$$= \frac{\sum_j \overbrace{\prod_{j' \neq j} z_{j'}(\mathbf{x})}^{b_j(\mathbf{x})} v_j w_{ji}(\mathbf{w}_j^T \mathbf{x})}{\underbrace{\prod_k z_k(\mathbf{x})}_{a(\mathbf{x})}} \tag{A.3}$$

$$= \sum_j \underbrace{\frac{b_j(\mathbf{x}) v_j}{a(\mathbf{x})}}_{c_j(\mathbf{x})} w_{ji} \mathbf{w}_j^T \mathbf{x} \tag{A.4}$$

$$= \left[\sum_j \frac{b_j(\mathbf{x}) v_j}{a(\mathbf{x})} w_{ji} \mathbf{w}_j\right]^T \mathbf{x} \tag{A.5}$$

$$\nabla_{\mathbf{x}} E = W C(\mathbf{x}) W^T \mathbf{x} \tag{A.6}$$

Here: $v_j > 0$; $b_j(\mathbf{x}) > 0$; $a_j(\mathbf{x}) > 0$ and therefore also $c_j(\mathbf{x}) > 0$. Also, $C = diag(c_1(\mathbf{x}), \ldots, c_N(\mathbf{x}))$ where M is the number of experts, $j = 1 \ldots M$. W is the matrix with the filters in its columns, i.e. the $D \times M$ dimensional matrix $W = (\mathbf{w}_1 \ldots \mathbf{w}_M)$. At the minimum we require $\nabla_{\mathbf{x}} E = \mathbf{0}$ and thus $\mathbf{0} = W C(\mathbf{x}) W^T \mathbf{x}$. Obviously, $\mathbf{x} = \mathbf{0}$ is one solution. Because

$c_j(\mathbf{x}) > 0 \;\; \forall j$ (in fact, c_j is monotonously increasing with the norm of $\mathbf{x}$) and because the matrix C is diagonal and therefore only scales the filter vectors in W, this is also the only solution as long as WW^T has full rank, i.e. W contains at least one subset of D linearly independent vectors $\mathbf{w}_j$.

For the FoE W consists of the M concatenated convolution matrices corresponding the different M experts (filters).

A.2 Mixture of Gaussian-BiFoE

The energy of the MoG-BiFoE

$$E_{\text{MoG}}(\mathbf{x}) = -\sum_i \sum_j \log \left\{ \exp\left[-\frac{1}{2} \left(\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j \right)^2 \right] + \exp\left[-\frac{1}{2} \left(\mathbf{w}_j^t \mathbf{x}_{(i)} - b_j - \Delta_j \right)^2 \right] \right\} \tag{A.7}$$

can be obtained as the free energy with respect to $\mathbf{x}$ in a model that includes a set of auxiliary variables z_{ij} which determine, for each expert j and image pixel i which of the two modes in the expert function is "active":

$$E_{\text{MoG}}^{\text{Aux}}(\mathbf{x}, \mathbf{z}) = \sum_{i,j} \left[\frac{z_{ij}}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j)^2 + \frac{(1 - z_{ij})}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j - \Delta_j)^2 \right] \tag{A.8}$$

(cf. equations 3.20 and 3.21 in the main text). This can be seen as follows:

$$\begin{aligned}
&\log \sum_{\mathbf{z}} \exp\left\{ -E_{\text{MoG}}^{\text{Aux}}(\mathbf{x}, \mathbf{z}) \right\} \\
&= \log \sum_{\mathbf{z}} \exp\left\{ -\sum_{i,j} \left[\frac{z_{ij}}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j)^2 + \frac{1 - z_{ij}}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j - \Delta_j)^2 \right] \right\} \\
&= \log \sum_{\mathbf{z}} \prod_{i,j} \exp\left[-\frac{z_{ij}}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j)^2 - \frac{(1 - z_{ij})}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j - \Delta_j)^2 \right] \\
&= \log \prod_{i,j} \sum_{z_{ij}} \exp\left[-\frac{z_{ij}}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j)^2 - \frac{(1 - z_{ij})}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j - \Delta_j)^2 \right] \\
&= \log \prod_{i,j} \left[\exp\left\{ -\frac{1}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j)^2 \right\} + \exp\left\{ -\frac{1}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j - \Delta_j)^2 \right\} \right] \\
&= \sum_{i,j} \log \left[\exp\left\{ -\frac{1}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j + \Delta_j)^2 \right\} + \exp\left\{ -\frac{1}{2} (\mathbf{w}_j^T \mathbf{x}_{(i)} - b_j - \Delta_j)^2 \right\} \right] \\
&= -E_{\text{MoG}}(\mathbf{x})
\end{aligned} \tag{A.9}$$

The conditional distribution of the auxiliary variables z_{ij} given an image $\mathbf{x}$ is given

as follows:

$$p(\mathbf{z}|\mathbf{x}) = \prod_{ij} p(z_{ij}|\mathbf{x}) \tag{A.10}$$

$$p(z_{ij}=1|\mathbf{x}) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{w}_j^T\mathbf{x}_{(i)} - b_j + \Delta_j)^2\right\}}{\exp\left\{-\frac{1}{2}(\mathbf{w}_j^T\mathbf{x}_{(i)} - b_j + \Delta_j)^2\right\} + \exp\left\{-\frac{1}{2}(\mathbf{w}_j^T\mathbf{x}_{(i)} - b_j - \Delta_j)^2\right\}} \tag{A.11}$$

A.3 Illustration of texture similarity scores

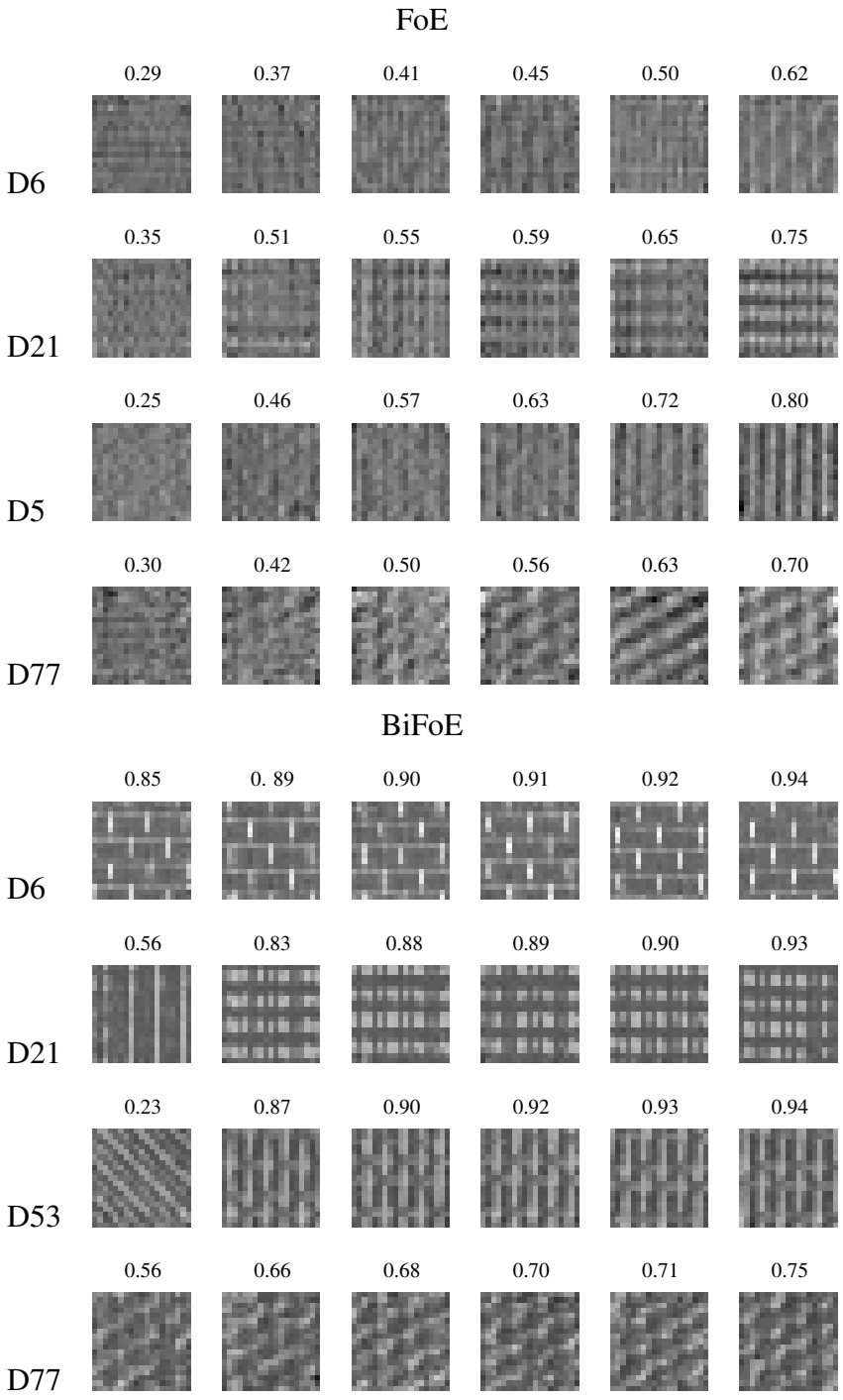
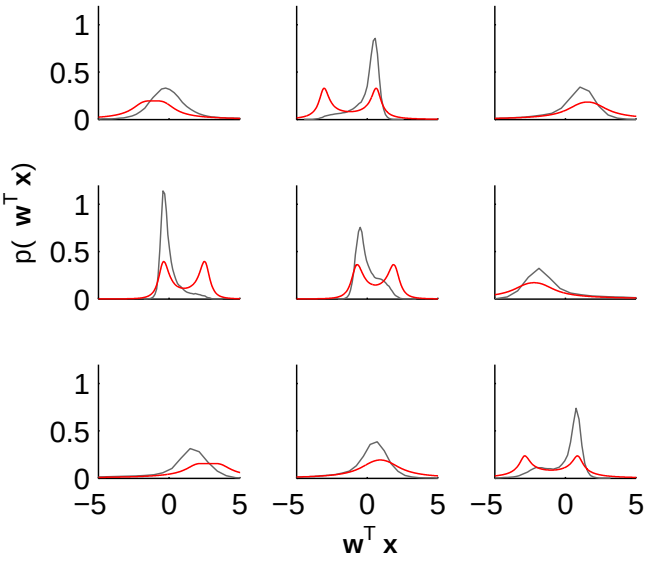
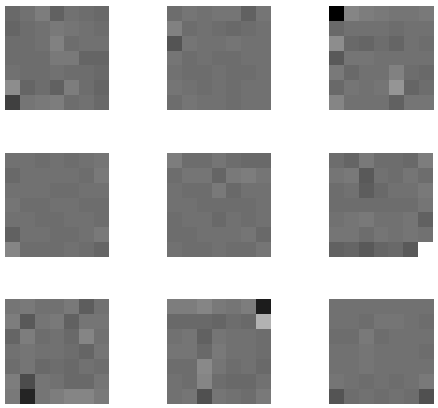


Figure A.1: Illustration of the correlation between similarity scores and visual quality of the samples. The figure shows examples of 19×19 samples from the FoE and BiFoE models with associated texture similarity scores.

A.4 Additional Model Parameters

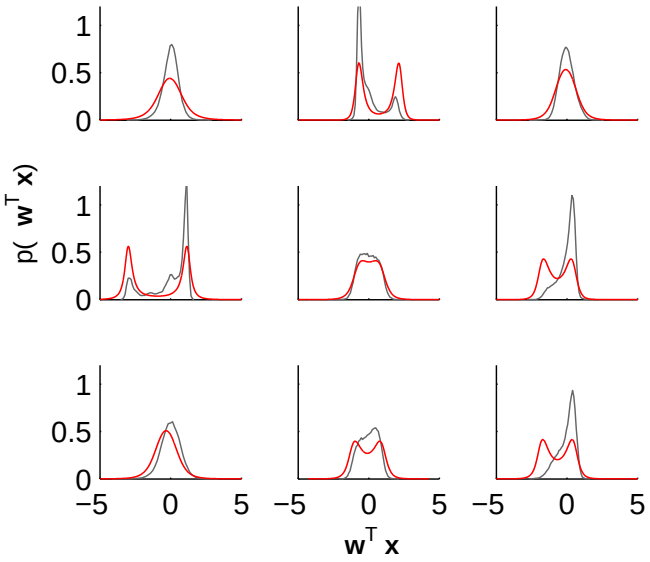


(a)

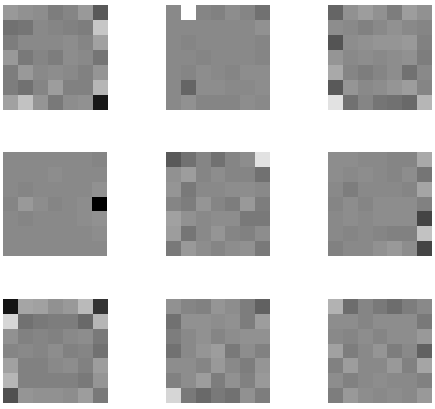


(b)

Figure A.2: Parameters of the BiFoE model for texture D6. *Top*: Expert nonlinearities for the nine experts used (*red*) and filter response marginals of the corresponding filters for the training data (*light gray*). *Bottom*: Corresponding filters.

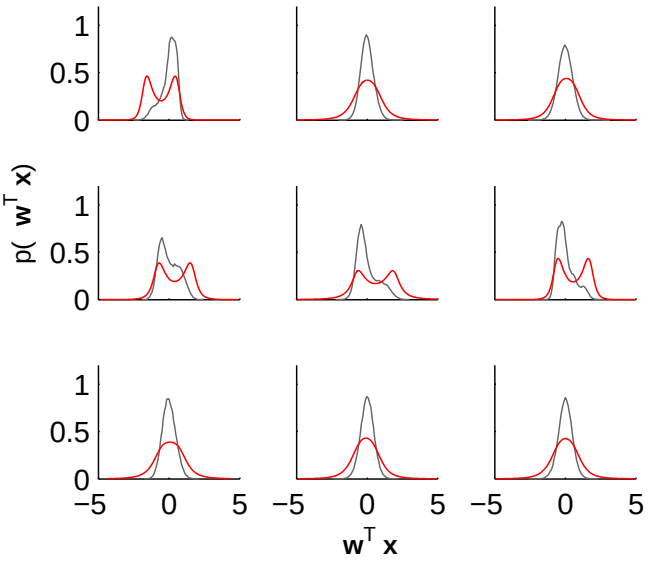


(a)

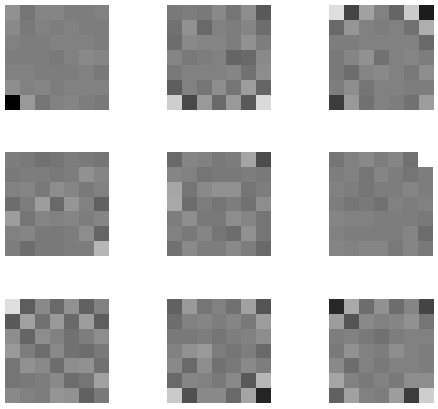


(b)

Figure A.3: Parameters of the BiFoE model for texture D21. *Top*: Expert nonlinearities for the nine experts used (*red*) and filter response marginals of the corresponding filters for the training data (*light gray*). *Bottom*: Corresponding filters.



(a)



(b)

Figure A.4: Parameters of the BiFoE model for texture D21. *Top*: Expert nonlinearities for the nine experts used (*red*) and filter response marginals of the corresponding filters for the training data (*light gray*). *Bottom*: Corresponding filters.

A.5 Inference in the hierarchical, region-based BiFoE

To see that inference in the hierarchical region-based BiFoE can indeed be performed as described in section 3.5.1 recall that the joint distribution over all variables involved, i.e. over the observed image $\mathbf{y}$, the mask $\mathbf{m}$, and the two latent images $\mathbf{x}_1$, and $\mathbf{x}_2$is given as

$$p(\mathbf{y},\mathbf{m},\mathbf{x}_1,\mathbf{x}_2) = p_M(\mathbf{m})p_1(\mathbf{x}_1)p_2(\mathbf{x}_2)\prod_{i=1}^{N}\delta(y_i,x_{1,i})^{m_i}\delta(y_i,x_{2,i})^{1-m_i}. \tag{A.12}$$

In general we have

$$p(m_i,x_{1i},x_{2i}|\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) = p(m_i|\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i})p(x_{1i},x_{2i}|\mathbf{y},m_i,\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}). \tag{A.13}$$

Furthermore:

$$p(m_i=1|\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) = \frac{p(m_i=1,\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i})}{p(\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i})} \tag{A.14}$$

where

$$p(\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) = p(m_i=1,\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) + p(m_i=0,\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}). \tag{A.15}$$

For $p(m_i=1,\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i})$ we have:

$$\begin{aligned} p(m_i=1,\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) &= \int \mathrm{d}x_{1i}\int \mathrm{d}x_{2i}\, p(m_i,x_{1i},x_{2i},\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) \\ &= p_M(m_i=1,\mathbf{m}_{\setminus i})p_1(y_i,\mathbf{x}_{1\setminus i})p_2(\mathbf{x}_{2\setminus i}) \\ &= p_M(m_i=1|\mathbf{m}_{\setminus i})p(\mathbf{m}_{\setminus i})p_1(y_i|\mathbf{x}_{1\setminus i})p_1(\mathbf{x}_{1\setminus i})p_2(\mathbf{x}_{2\setminus i}) \end{aligned} \tag{A.16}$$

and similarly

$$p(m_i=0,\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) = p_M(m_i=0|\mathbf{m}_{\setminus i})p(\mathbf{m}_{\setminus i})p_2(y_i|\mathbf{x}_{2\setminus i})p_2(\mathbf{x}_{2\setminus i})p_1(\mathbf{x}_{1\setminus i}), \tag{A.17}$$

so that we have for eq. (A.14)

$$\begin{aligned} &p(m_i=1|\mathbf{y},\mathbf{m}_{\setminus i},\mathbf{x}_{1\setminus i},\mathbf{x}_{2\setminus i}) = \\ &\frac{p_M(m_i=1|\mathbf{m}_{\setminus i})p(\mathbf{m}_{\setminus i})p_1(y_i|\mathbf{x}_{1\setminus i})p_1(\mathbf{x}_{1\setminus i})p_2(\mathbf{x}_{2\setminus i})}{\left(p_M(m_i=1|\mathbf{m}_{\setminus i})p_1(y_i|\mathbf{x}_{1\setminus i}) + p_M(m_i=0|\mathbf{m}_{\setminus i})p_2(y_i|\mathbf{x}_{2\setminus i})\right)p(\mathbf{m}_{\setminus i})p_1(\mathbf{x}_{1\setminus i})p_2(\mathbf{x}_{2\setminus i})} \\ &= \frac{p_M(m_i=1|\mathbf{m}_{\setminus i})p_1(y_i|\mathbf{x}_{1\setminus i})}{p_M(m_i=1|\mathbf{m}_{\setminus i})p_1(y_i|\mathbf{x}_{1\setminus i}) + p_M(m_i=0|\mathbf{m}_{\setminus i})p_2(y_i|\mathbf{x}_{2\setminus i})} \end{aligned} \tag{A.18}$$

as in equation (3.26) in the main text. Equation (3.27) then follows directly from the definition of the joint distribution in eq. (A.12).

Appendix B

Masked RBM

B.1 Gibbs sampling scheme for the masked RBM with uniform shape model

For the masked RBM with uniform shape model we obtain the following joint distribution (cf. also eq. 4.1 in the main text):

$$P(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K} | \mathbf{m}) \propto \left(\prod_i \delta\left[\widehat{v}_{m_i,i} = v_i\right] \right) \left(\prod_k \mathrm{APP}(\widehat{\mathbf{v}}_k, \mathbf{h}^{(a)}_k) \right). \tag{B.1}$$

This joint distribution exhibits several properties:

1. given the latent images $\widehat{\mathbf{v}}_{1..K}$, the distribution over the appearance hidden states $\mathbf{h}^{(a)}_{1..K}$ is factorial (APP is an RBM)

2. given the image patch $\mathbf{v}$ and the hidden states $\mathbf{h}^{(a)}_{1..K}$ the conditional distribution over the mask $\mathbf{m}$ is factorial (see eq. 4.3 in the main text)

3. given the image patch $\mathbf{v}$, the mask $\mathbf{m}$ and the hidden states $\mathbf{h}^{(a)}_k$, the distribution over the latent images $\widehat{\mathbf{v}}_k$ is factorial (again, see eq. 4.3 in the main text).

These properties suggest the following Gibbs sampling scheme to infer all the hidden variables given an image $\mathbf{v}$: starting from a random mask $\mathbf{m}$, we iterate over the following steps:

1. given the mask $\mathbf{m}$, we sample the unobserved parts of the latent images $\widehat{\mathbf{v}}_{1..K}$ using block Gibbs sampling (using properties 1 and 3)

2. given the (completed) latent images $\widehat{\mathbf{v}}_{1..K}$, we sample the appearance hidden units $\mathbf{h}^{(a)}_{1..K}$ (using property 1)

3. given the appearance hidden units $\mathbf{h}^{(a)}_{1..K}$ and the image patch $\mathbf{v}$ we sample a new mask $\mathbf{m}$ (using property 2)

This process is repeated until convergence of the mask. The sampling procedure directly implies that the mask may be different each time. However, in all experiments, it typically matched the structure of the shapes in the images.

B.2 Conditional distribution of mask is factorial given hiddens

Below we show that given a depth order π and the state of the shape hidden units $\mathbf{h}^{(s)}_{1...K}$ the conditional distribution over the mask is factorial, i.e. that

$$\begin{aligned} P(m_i = k|\mathbf{h}^{(s)}_{1..K},\pi) \quad &\propto \quad \text{SHAPE}(s_{k,i} = 1|\mathbf{h}^{(s)}_k) \\ &\quad \times \prod_{k':\pi(k')<\pi(k)} \left[1 - \text{SHAPE}(s_{k',i} = 1|\mathbf{h}^{\prime(s)}_k)\right]. \end{aligned}$$

(cf. equation 4.16 in the main text). Specifically we will show that in

$$p(\mathbf{m}|\mathbf{h}_{1...K},\pi) = \frac{\sum_{\mathbf{s}_{1...K}} p(\mathbf{m},\mathbf{s}_{1...K},\mathbf{h}_{1...K}|\pi)}{\sum_{\mathbf{m}}\sum_{\mathbf{s}_{1...K}} p(\mathbf{m},\mathbf{s}_{1...K},\mathbf{h}_{1...K}|\pi)} \tag{B.2}$$

the numerator as well as the denominator factor with respect to the pixels. The derivation below is shown for the case when shapes are modeled in all layers but it holds in the same way if the rear-most layer is assumed to have a fixed (all on) shape as discussed in section 4.3.2.

For the proof we make use of the fact that the joint distribution over a visible and hidden units of any RBM can be written as a product of experts hidden units either with respect to the visible units or with respect to the hidden units (see also chapter 2, equation (2.28) in section 2.2.4). Here we make use of the former, i.e. we will write the joint distribution over shape visible and hidden units as follows:

$$\text{SHAPE}(\mathbf{s},\mathbf{h}) = \frac{1}{Z}\prod_i f(s_i,\mathbf{h}), \tag{B.3}$$

where Z is the normalization constant of the RBM. We also use

$$Z_i(\mathbf{h}) = f(s_i = 1,\mathbf{h}) + f(s_i = 0,\mathbf{h}). \tag{B.4}$$

Using this property we can re-write equation (4.12) in the main text as follows:

$$p(\mathbf{m},\mathbf{s}_{1\ldots K},\mathbf{h}_{1\ldots K}|\pi) = \prod_k \text{SHAPE}(\mathbf{s}_k,\mathbf{h}_k)\prod_i \delta(s_{m_i,i}=1)\prod_{k:\pi(k)<\pi(m_i)}\delta(s_{k,i}=0) \tag{B.5}$$

$$= \frac{1}{Z^K}\prod_i\left(\prod_k f(s_{k,i},\mathbf{h}_k)\delta(s_{m_i,i}=1)\prod_{k:\pi(k)<\pi(m_i)}\delta(s_{k,i}=0)\right) \tag{B.6}$$

Now we can consider the denominator of equation (B.2):

$$\sum_{\mathbf{m}}\sum_{\mathbf{s}_{1\ldots K}} p(\mathbf{m},\mathbf{s}_{1\ldots K},\mathbf{h}_{1\ldots K}|\pi) \tag{B.7}$$

$$= \frac{1}{Z^K}\prod_i\sum_{m_i}\left(f(s_i=1,\mathbf{h}_{m_i})\prod_{k:\pi(k)<\pi(m_i)} f(s_i=0,\mathbf{h}_k)\prod_{k:\pi(k)>\pi(m_i)}\sum_{s_i} f(s_i,\mathbf{h}_k)\right) \tag{B.8}$$

$$= \frac{1}{Z^K}\prod_i\sum_{m_i}\left(\prod_{k:\pi(k)<\pi(m_i)} Z_i(\mathbf{h}_k)\text{SHAPE}(s_i=0|\mathbf{h}_k)\right.$$
$$\left.\times Z_i(\mathbf{h}_{m_i})\text{SHAPE}(s_i=1|\mathbf{h}_{m_i})\prod_{k:\pi(k)>\pi(m_i)} Z_i(\mathbf{h}_k)\right) \tag{B.9}$$

$$= \frac{1}{Z^K}\prod_i\prod_k Z_i(\mathbf{h}_k)\sum_{m_i}\left(\text{SHAPE}(s_i=1|\mathbf{h}_{m_i})\prod_{k:\pi(k)<\pi(m_i)}[1-\text{SHAPE}(s_i=1|\mathbf{h}_k)]\right), \tag{B.10}$$

where we have used that $\text{SHAPE}(s_i|\mathbf{h}) = f(s_i,\mathbf{h})/Z_i(\mathbf{h})$. Using this we find for the full expression in equation (B.2):

$$p(\mathbf{m}|\mathbf{h}_{1\ldots K},\pi) = \frac{\sum_{\mathbf{s}_{1\ldots K}} p(\mathbf{m},\mathbf{s}_{1\ldots K},\mathbf{h}_{1\ldots K}|\pi)}{\sum_{\mathbf{m}}\sum_{\mathbf{s}_{1\ldots K}} p(\mathbf{m},\mathbf{s}_{1\ldots K},\mathbf{h}_{1\ldots K}|\pi)}$$

$$= \frac{\frac{1}{Z^K}\prod_i\prod_k Z_i(\mathbf{h}_k)\prod_{k:\pi(k)<\pi(m_i)}[1-\text{SHAPE}(s_i=1|\mathbf{h}_k)]\text{SHAPE}(s_i=1|\mathbf{h}_{m_i})}{\frac{1}{Z^K}\prod_i\prod_k Z_i(\mathbf{h}_k)\sum_{m'_i}\prod_{k:\pi(k)<\pi(m'_i)}[1-\text{SHAPE}(s_i=1|\mathbf{h}_k)]\text{SHAPE}(s_i=1|\mathbf{h}_{m'_i})}$$

$$= \prod_i \frac{\prod_{k:\pi(k)<\pi(m_i)}[1-\text{SHAPE}(s_i=1|\mathbf{h}_k)]\text{SHAPE}(s_i=1|\mathbf{h}_{m_i})}{\sum_{m'_i}\prod_{k:\pi(k)<\pi(m'_i)}[1-\text{SHAPE}(s_i=1|\mathbf{h}_k)]\text{SHAPE}(s_i=1|\mathbf{h}_{m'_i})} \tag{B.11}$$

$$\propto \prod_i\prod_{k:\pi(k)<\pi(m_i)}[1-\text{SHAPE}(s_i=1|\mathbf{h}_k)]\text{SHAPE}(s_i=1|\mathbf{h}_{m_i}) \tag{B.12}$$

Therefore

$$p(\mathbf{m}|\mathbf{h}_{1\ldots K},\pi) = \prod_i p(m_i|\mathbf{h}_{1\ldots K},\pi) \tag{B.13}$$

$$p(m_i|\mathbf{h}_{1\ldots K},\pi) \propto \text{SHAPE}(s_i = 1|\mathbf{h}_{m_i}) \prod_{k:\pi(k)<\pi(m_i)} [1 - \text{SHAPE}(s_i = 1|\mathbf{h}_k)]. \tag{B.14}$$

Note that if we treat the rear-most layer in a special manner and assume its shape to be on everywhere (see discussion in section 4.3.2) then we have for equation (B.11)

$$p(\mathbf{m}|\mathbf{h}_{1\ldots K},\pi) \tag{B.15}$$

$$= \prod_i \frac{\prod_{k:\pi(k)<\pi(m_i)} [1 - \text{SHAPE}(s_i = 1|\mathbf{h}_k)] [\text{SHAPE}(s_i = 1|\mathbf{h}_{m_i})]^{\delta(\pi(m_i)<K)}}{\sum_{m'_i} \prod_{k:\pi(k)<\pi(m'_i)} [1 - \text{SHAPE}(s_i = 1|\mathbf{h}_k)] \left[\text{SHAPE}(s_i = 1|\mathbf{h}_{m'_i})\right]^{\delta(\pi(m'_i)<K)}}$$

$$= \prod_i \prod_{k:\pi(k)<\pi(m_i)} [1 - \text{SHAPE}(s_i = 1|\mathbf{h}_k)] [\text{SHAPE}(s_i = 1|\mathbf{h}_{m_i})]^{\delta(\pi(m_i)<K)}, \tag{B.16}$$

i.e. we have an equality instead of a proportionality. This last equality holds since

$$\sum_{m_i} \prod_{k:\pi(k)<\pi(m'_i)} [1 - \text{SHAPE}(s_i = 1|\mathbf{h}_k)] \left[\text{SHAPE}(s_i = 1|\mathbf{h}_{m'_i})\right]^{\delta(\pi(m_i)<K)} = 1. \tag{B.17}$$

Appendix C

Field of masked RBMs

C.1 Alternative formulation of the occlusion model

In section 5.1.2 in the main text we describe an (approximate) generative process for the occlusion model in the Field of masked RBMs that is given by equation (5.2): We sample the shapes independently, and then for each image pixel, we force the corresponding shape pixel of the rear-most patch to be on if that image pixel would otherwise not be explained by any latent patch. This avoids having to sample all shapes jointly.

One way to think about this generative process is that when we generate the mask we effectively ignore the state of the rear-most shape pixel. This can be interpreted in terms of the following well-defined generative model:

$$P(\mathbf{m}, \mathbf{s}_{1..L}, \mathbf{h}^{(s)}_{1..L}, \pi) = P(\pi) \left(\prod_i \delta(s_{m_i, r_{m_i}(i)} = 1)^{\delta(\text{patch } m_i \text{ is not rear-most at } i)} \prod_{l \in o(i): \pi(l) < \pi(m_i)} \delta(s_{l, r_l(i)} = 0) \right) \times \left(\prod_l \text{SHAPE}(\mathbf{s}_l, \mathbf{h}^{(s)}_l) \right). \quad \text{(C.1)}$$

Note that in this formulation, for m_i to take value l the corresponding shape pixel of the latent patch only has to be on if l is *not* the rear-most patch at that image pixel (the first constraint in equation (C.1) is active only if the patch is not the rear-most patch at pixel i). This is in contrast to the model defined by equation (5.2) where the corresponding shape pixel always has to be on.

This model directly reflects the generative process used in practice but has the undesirable property the rear-most latent patch at a particular pixel can be visible at that

pixel even though its shape suggests that it should not be. The model that we describe in the Discussion of chapter 5 (section 5.4.1.1) does not suffer from this problem but still achieves marginal independence of the shapes.

One consequence of this property is that given a mask $\mathbf{m}$ and a depth ordering π we have a larger set of unobserved shape pixels: Normally, the set of unobserved shape pixels contains all those shape pixels that are occluded by other shapes as illustrated in Figure 5.4 in the main text (see also Fig. 4.6 for a similar illustration for the masked RBM). In the above formulation of the model, however, the set of unobserved shape pixels also contains those pixels of a shape that are visible (i.e. they are not occluded by some other shape in front) but where the corresponding patch is rear-most.

Depth inference: Exact depth inference in the model defined by (C.1) requires treating all rear-most shape pixels (i.e. all those pixels for which a particular patch is the rear-most patch) as unobserved even when the mask suggests that the latent patch is visible at that pixel. Eq. (5.3) then remains the same, just the set of pixels that need to be filled in to obtain the completed latent shape $\hat{\mathbf{s}}^{\pi'}_{l_n}$ is different (all those pixels for which the latent patch is rear-most are included in this set). Given a mask the relative probabiblity of a particular depth order can thus be approxmiated using a set of partially observed and completed shapes that are consistent with that depth order using equation (5.3) given in the main text which we provide here for completeness:

$$P(\pi'|\mathbf{m},\hat{\mathbf{s}}^{\pi'}_{1..K},\pi) \propto \prod_{l:l\neq l_0} \delta(\pi'(l)=\pi(l)) \prod_{n=0}^{N} \text{SHAPE}(\hat{\mathbf{s}}^{\pi'}_{l_n}) \tag{C.2}$$

Mask inference: Equation (5.6) in the main text, which is reproduced below, is the exact Gibbs sampling step for this model:

$$P(m_i = l|\mathbf{h}^{\prime(s)}_{1..L},\pi') = \begin{cases} \displaystyle\prod_{l'\in o(i):\pi(l')<\pi(l)} \text{SHAPE}(s_{l',r_{l'}(i)}=0|\mathbf{h}^{\prime(s)}_{l'}) & \text{if } l \text{ is rear-most} \\ & \text{patch at pixel } i \\ \text{SHAPE}(s_{l,r_l(i)}=1|\mathbf{h}^{\prime(s)}_{l}) \times & \\ \displaystyle\prod_{l'\in o(i):\pi(l')<\pi(l)} \text{SHAPE}(s_{l',r_{l'}(i)}=0|\mathbf{h}^{\prime(s)}_{l'}) & \text{otherwise,} \end{cases} \tag{C.3}$$

Learning: Learning for this model proceeds as described in the main text (cf. section 4.3.3.2), except for the fact that the set of unobserved shape pixels changes: All those shape pixels of a patch for which that patch is the rear-most are also unobserved and need to be sampled from the posterior. Note that as for the masked RBM the independence assumption made by the CD update is now correct.