High-Order Markov Random Fields for Low-Level Vision

by
Stefan Roth
Dipl. Inf., University of Mannheim, Germany, 2001
Sc. M., Brown University, 2003

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy in the
Department of Computer Science at Brown University

Providence, Rhode Island
May 2007

This dissertation by Stefan Roth is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ________________ ____________________________________

Michael J. Black, Director

Recommended to the Graduate Council

Date ________________ ____________________________________

John F. Hughes, Reader

Date ________________ ____________________________________

Yair Weiss, Reader
(Hebrew University, Jerusalem)

Approved by the Graduate Council

Date ________________ ____________________________________

Sheila Bonde
Dean of the Graduate School

ABSTRACT

Low-level vision is a fundamental area of computer vision that is concerned with the analysis of digital images at the pixel level and the computation of other dense, pixel-based representations of scenes such as depth and motion. Many of the algorithms and models in low-level vision rely on a representation of prior knowledge about images or other dense scene representations. In the case of images, this prior knowledge represents our *a-priori* belief in observing a particular image among all conceivable images. Such prior knowledge can be supplied in a variety of different ways; a wide range of low-level vision techniques represent the prior belief using Markov random fields (MRFs). MRFs are a compact and efficient probabilistic representation, and are particularly appropriate for spatially arranged data, such as the pixels in an image. Markov random fields have a long history in low-level computer vision; their representational power, however, has often been limited by restricting them to very local spatial structures.

This dissertation introduces a novel, expressive Markov random field model for representing prior knowledge in low-level vision, for example about images and image motion (optical flow). This high-order MRF model, called *Fields of Experts* (FoE), represents interactions over larger spatial neighborhoods compared to many previous MRF models. Learning the parameters of large MRF models from training data, as well as inferring the quantity of interest (e. g., the noise-free image) are known to be very challenging, both algorithmically and computationally. This is even more so in models that represent complex spatial interactions and have many parameters, such as the FoE model. This dissertation describes machine learning techniques that enable approximate learning and inference with these models. The core thesis developed in this work is that these high-order Markov random fields are more powerful models for representing prior knowledge in low-level vision than previous MRF models, and that they lead to competitive algorithms for varied problems such as image denoising and the estimation of image motion.

VITA

Stefan Roth was born on March 13, 1977 in Mainz, Germany.

Education

- *Ph.D. in Computer Science*, Brown University, Providence, RI, USA, May 2007.
- *Sc.M. in Computer Science*, Brown University, Providence, RI, USA, May 2003.
- *Diplom in Technische Informatik (Computer Science and Engineering)*, University of Mannheim, Germany, May 2001.

Honors

- *Marr Prize (Honorable Mention)*, October 2005, 10th IEEE International Conference on Computer Vision for the paper "On the spatial statistics of optical flow" (jointly with Michael J. Black).
- *Sigma Xi Outstanding Graduate Student Research Award*, April 2005.
- *Sigma Xi*, elected Associate Member, April 2005.
- *Dean's Fellowship*, Brown University, 2001-2002 academic year.
- *Fulbright Travel Grant*, German/American Fulbright Association, 2001-2006.

Academic Experience

- Research Assistant, Department of Computer Science, Brown University, 2002-2007.
- Teaching Assistant for Computer Science 0143 (Introduction to Computer Vision), Brown University, Fall 2003.
- Undergraduate Research Assistant, Department of Mathematics and Computer Science, University of Mannheim, Germany, 1997-1998.

Professional Experience

- Intern at Intel Research, Santa Clara, CA, Summers 2003 & 2004.
- Intern at Mitsubishi Electric Research Laboratory, Cambridge, MA, Spring 2000.
- Freelance Programmer for Volume Graphics GmbH, Heidelberg, Germany, 1998-2001.

Peer-Reviewed Journal Papers

- Christian Schellewald, Stefan Roth, and Christoph Schnörr. Evaluation of a convex relaxation to a quadratic assignment matching approach for relational object views. *Image and Vision Computing*, 2007. doi:10.1016/j.imavis.2006.08.005. To appear.
- Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50, August 2007. doi:10.1007/s11263-006-0016-x.

Peer-Reviewed Conference Papers

- Teodor Mihai Moldovan, Stefan Roth, and Michael J. Black. Denoising archival films using a learned Bayesian model. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2641–2644, Atlanta, Georgia, October 2006. doi:10.1109/ICIP.2006.313052.
- Stefan Roth and Michael J. Black. Specular flow and the recovery of surface structure. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1869–1876, New York, New York, June 2006. doi:10.1109/CVPR.2006.290.
- Xiangyang Lan, Stefan Roth, Daniel P. Huttenlocher, and Michael J. Black. Efficient belief propagation with learned higher-order Markov random fields. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proceedings of the Ninth European Conference on Computer Vision*, volume 3952 of *Lecture Notes in Computer Science*, pages 269–282. Springer, 2006. doi:10.1007/11744047_21.
- Frank Wood, Stefan Roth, and Michael J. Black. Modeling neural population spiking activity with Gibbs distributions. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1539–1546, 2006.
- Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 1,

pages 42–49, Beijing, China, October 2005b. doi:10.1109/ICCV.2005.180. *Oral presentation.*

- Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, San Diego, California, June 2005a. doi:10.1109/CVPR.2005.160. *Oral presentation.*

- Stefan Roth, Leonid Sigal, and Michael J. Black. Gibbs likelihoods for Bayesian tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, Washington, DC, June 2004. doi:10.1109/CVPR.2004.116.

- Leonid Sigal, Sidharth Bhatia, Stefan Roth, Michael J. Black, and Michael Isard. Tracking loose-limbed people. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 421–428, Washington, DC, June 2004. doi:10.1109/CVPR.2004.252.

- Christian Schellewald, Stefan Roth, and Christoph Schnörr. Evaluation of convex optimization techniques for the weighted graph-matching problem in computer vision. In B. Radig and S. Florczyk, editors, *Pattern Recognition, Proceedings of the 23rd DAGM-Symposium*, volume 2191 of *Lecture Notes in Computer Science*, pages 361–368. Springer, 2001.

- H.-J. Bender, R. Männer, C. Poliwoda, S. Roth, and M. Walz. Reconstruction of 3D catheter paths from 2D x-ray projections. In C. Taylor and A. C. F. Colchester, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI'99*, volume 1679 of *Lecture Notes in Computer Science*, pages 981–989. Springer, 1999. doi:10.1007/10704282_107.

Undergraduate Thesis

- Stefan Roth. Analysis of a deterministic annealing method for graph matching and quadratic assignment problems in computer vision. Diplom thesis, University of Mannheim, Germany, May 2001.

Technical Reports

- Christian Schellewald, Stefan Roth, and Christoph Schnörr. Performance evaluation of a convex relaxation approach to the quadratic assignment of relational object views. Technical Report TR-2002-02, University of Mannheim, Germany, February 2002.

Refereed Abstracts

- Michael J. Black and Stefan Roth. On the receptive fields of Markov random fields. Cosyne, 2005.
- Stefan Roth, Fulvio Domini, and Michael J. Black. Specular flow and the perception of surface reflectance. *Journal of Vision*, 3(9):413a, 2003. doi:10.1167/3.9.413.

ACKNOWLEDGMENTS

This dissertation would not have been possible without the help and support of a great number of people. Without them, the journey to the point where I am writing this would not have been half as fun and rewarding.

First and foremost, I wish to thank my advisor, Michael Black. He has been a terrific teacher, role model, and friend. Over the years I have spent here, Michael's research approach and style have shaped me a lot, and I have learned a great deal from him. He taught me how to find and approach interesting research problems, how to find my own research style, how to be a good scholar, how to present my work to the scientific community, how to make complex problems accessible to others, and many other things; too many to mention here. But not only that, he has also provided invaluable support and advice on teaching, advising students, finding jobs, and on life in general. Even during tougher times, his passion for research and his belief in my abilities were the key for staying motivated and focused. I am still amazed by his drive and his curiousness, and I sincerely hope that I picked up some of that along the way. Because of that, much of the credit for what I have accomplished over the past almost six years goes to Michael. Thank you! It has been a great time.

I would also like to thank my thesis committee members, John Hughes and Yair Weiss, for their help and support both with my dissertation and beyond. Their critical questions, especially about my thesis proposal, have guided me in the right direction and are directly reflected in this dissertation. Their comments on draft versions were also very helpful and improved the document substantially.

Before coming to Brown, my interest in academia, computer science research, and computer vision was shaped by a few important people, whom I would like to thank here. My interest in computer science research was stimulated by Reinhard Männer, who gave me the opportunity to work as an undergraduate research assistant in his lab, and later paved the way toward spending three very interesting and important months as an intern at MERL. My internship advisor there, Sarah Frisken-Gibson, helped me tremendously by shaping and focusing my ideas about graduate studies. Finally, my undergraduate advisor Christoph Schnörr furthered my interest in computer vision and machine learning, both during his classes as well as during the enjoyable time that I spent in his group writing my undergraduate thesis.

During my PhD studies I spent two great summers as an intern at Intel Research, for which I would like to thank Horst Haussecker and his group. They provided a stimulating and very pleasant environment that allowed me to grow my interest in image modeling, Markov random fields, and low-level vision problems in general.

Over the years of my PhD, I was fortunate to have several collaborators with whom I worked on a number of interesting projects, some of which led to or are even part of this dissertation. I would like to thank Sid Bhatia, Dan Huttenlocher, Michael Isard, Xiangyang Lan, Teodor Mihai Moldovan, Leonid Sigal, and Frank Wood for their good collaborations.

While at Brown, I have had many interesting research discussions with people who have not only provided many ideas, suggestions, and asked important questions that put me on the right track, but also became friends. I am grateful to Chad Jenkins, Greg Shakhnarovich, and Fernando de la Torre for that. I particularly wish to thank my "academic brother" and Bayesian accomplice Frank Wood. Not only did we have many inspiring research discussions over the years that I am very grateful for, but Frank always made sure that I did not forget the lighter side of things.

Quite a number of friends from Brown University and elsewhere have provided both invaluable encouragement and the necessary distraction, both of which made this ride more enjoyable. I would especially like to thank Yanif Ahmad, Yasemin Altun, Ionuţ Aron, Glencora Borradaile, Olga Karpenko, Amy Koo, Tomer Moscovich, Olga Papaemmanouil, Peter Sibley, Nikos Triandopoulos, and the whole Brown CS softball team. Thanks also to my long term friends back home in Germany, particularly to Oliver Kargus, Alexander Klee, Marc Mauritz, Lars Moser, Nina Rauth, and Christian Rauth. Whenever I visited Germany, they made it seem like I had not even been away.

Other than those already mentioned, a number of people have contributed to this work, either through helpful discussions and suggestions, by providing details on their work, or by making data available to me. I would like to thank Stuart Andrews, John Barron, Pietro Berkes, Marcelo Bertalmío, Andrew Blake, Andrés Bruhn, David Capel, Jay Cassidy, Alessandro Duci, Andrew Fitzgibbon, Yoram Gat, Stuart Geman, Thomas Hofmann, Siwei Lyu, Kevin Manbeck, Étienne Mémin, Oscar Nestares, Javier Portilla, Guillermo Sapiro, Hanno Scharr, Eero Simoncelli, Max Welling, and Andrew Zisserman for their help.

Finally, I am eternally grateful to my family, particularly to my parents, for their continued support of my ideas and plans, and their encouragement both before and during my graduate studies. Without them I probably would not have come to Brown and I would not have written this.

A scientist who has learned how to use probability theory directly as extended logic has a great advantage in power and versatility over one who has learned only a collection of unrelated ad hoc *devices.*

E. T. Jaynes, from *Probability Theory: The Logic of Science*

To my parents.

TABLE OF CONTENTS

⋆ Parts of this dissertation are joint work with other authors and have previously appeared in [Roth and Black, 2005a,b; Moldovan et al., 2006; Roth and Black, 2007].

LIST OF TABLES

LIST OF ILLUSTRATIONS

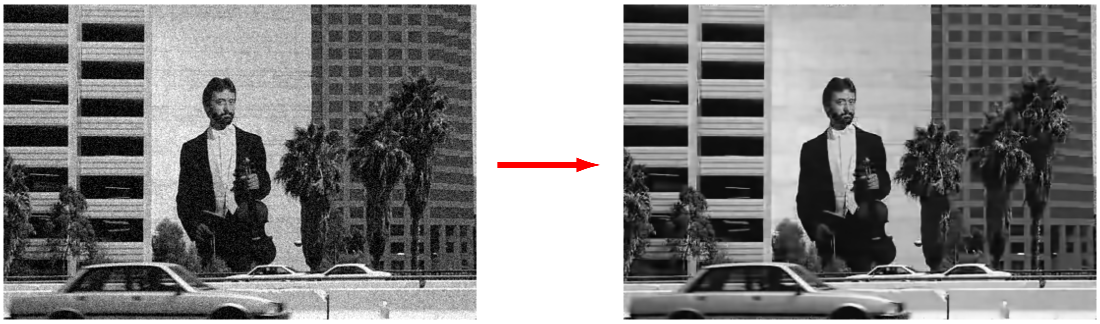
CHAPTER 1

Introduction

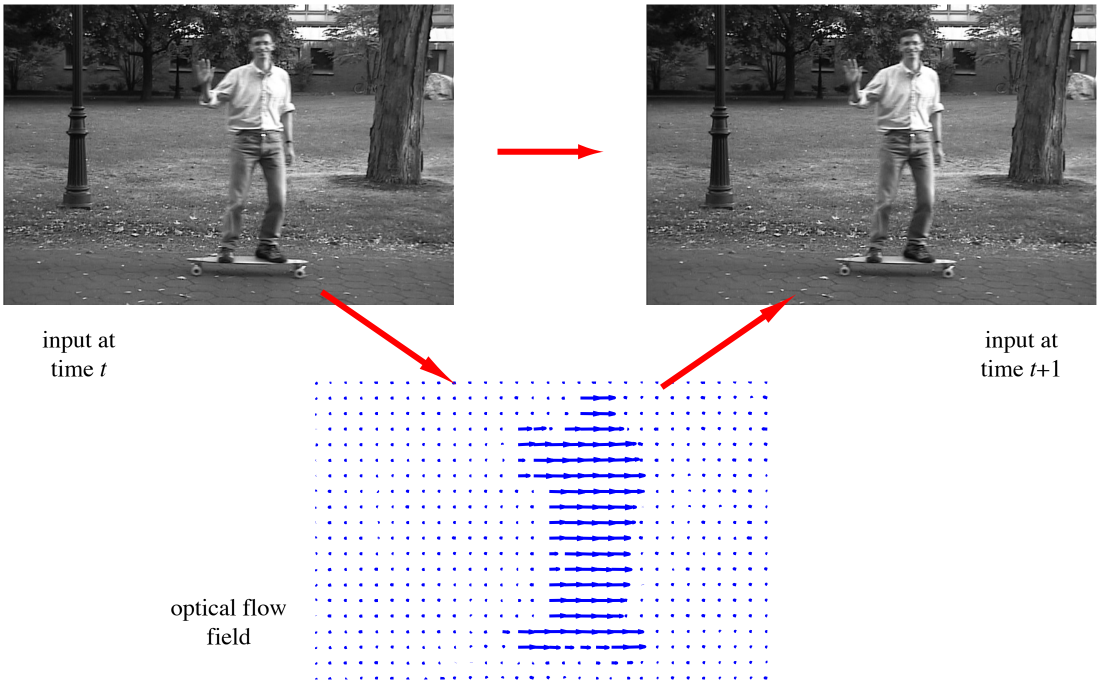
Computer or machine vision is a growing field concerned with understanding and interpreting digital images with computer algorithms [Marr, 1982; Horn, 1986]. Its problems and applications span a large spectrum of levels of abstractions, which ranges from extracting per-pixel information, also called low-level vision, all the way to a semantic understanding of the image content, also called high-level vision. This dissertation is focused on problems of low-level vision, such as image denoising [Rosenfeld and Kak, 1982], image motion estimation from image sequences (optical flow) [Horn and Schunck, 1981], or scene depth estimation from stereo image pairs [Marr and Poggio, 1976].

These applications are of substantial practical interest, and even though problems in low-level vision have been studied for decades, they have remained difficult: Many digital images are degraded by noise, for example by sensor noise in low light conditions. Digital cameras have found such widespread use in the consumer market that most of us will have encountered the problem of visible noise in images. In image denoising (cf. Fig. 1.1(a)) the task is to remove this noise and recover the image that would be observed if the imaging conditions were ideal. In image motion estimation, also called optical flow estimation (cf. Fig. 1.1(b)), the goal is to estimate the motion of each pixel between subsequent time steps (frames) in an image sequence. Optical flow has a wide range of applications, within and outside of computer vision; recent uses include special effects for movie productions and biological studies of plant growth. The problem of optical flow estimation is made difficult by image noise as well, but also by global and local lighting changes. More importantly, if there is a lack of image structure in parts of the observed image, then the image motion is locally ambiguous and thus difficult to determine. Similar difficulties exist in a large number of other machine vision and graphics problems including image inpainting, stereo, super-resolution, image-based rendering, volumetric surface reconstruction, and texture synthesis to name a few.

All these applications have in common that (1) they require estimating information from



(a) Image denoising. The noisy input is shown on the left, the denoised image on the right.



(b) Optical flow estimation.

Figure 1.1: **Two low-level vision problems.** In image denoising as shown in (a) the goal is to "invert" the process that leads to visible noise (such as sensor noise) in the image. In optical flow estimation as shown in (b) the goal is to recover the apparent motion of each pixel between two subsequent video frames (i. e., the motion field).

uncertain “noisy” sensor measurements; (2) our models of them are incomplete as they do not capture the full complexity of the image formation process[1]; and that (3) the available data is often insufficient to constrain all aspects of our solution. This makes many low-level vision problems mathematically ill-posed [Hadamard, 1923]. Because of that, it has long been recognized [e. g., Poggio et al., 1985] that in order to solve such low-level vision problems, we need to impose some kind of *prior knowledge* that constrains the space of possible solutions. This is commonly referred to as *regularization* [Tikhonov, 1963].

Let us consider a more concrete example in order to illustrate this: Suppose we cannot compute image motion in a part of the image due to the lack of local image structure. Then we could assume that the motion in this part is similar to the motion in neighboring regions (where it was possible to compute it) and disambiguate the problem this way. In other terms, one possible regularization method is to assume spatial smoothness, here of the recovered flow field. Similarly, in image denoising we can make use of the fact that nearby pixels in an image often have very similar intensity values (as discussed in more detail below), and assume spatial smoothness of the intensity values. Such very simple models of prior knowledge are too restricted, however, and do not lead to good results in practice. More sophisticated regularization techniques have been developed over the years (as reviewed in Section 2.2), but rich models of prior knowledge in low-level vision that express the complicated kinds of structures found, for example, in natural images have been mostly lacking so far. Moreover, to achieve very good performance on a specific application, many models of prior knowledge have been specialized so much that it is difficult to apply them to a different task. The main goal of this dissertation is to introduce a unifying framework that enables richer models of prior knowledge in low-level vision, and to show how these models help improve application performance across a varied range of applications.

To do so we will adopt a probabilistic formulation for low-level vision problems, which has many advantages including that the inherent uncertainty and ambiguity that we are faced with can be treated very naturally [e. g., Marroquin et al., 1987]. One very important aspect of this is that it allows training the model (i. e., choosing its parameters) from example data in a principled way. More specifically, we will adopt a Bayesian approach [Szeliski, 1990; Jaynes, 2003]. The Bayesian approach is characterized by the assumption that not only our measured data (e. g., the noisy image in denoising) is uncertain, but also the ideal, true state that gave rise to our measurement (e. g., the noise free image that we are trying to find). This is formalized using a so-called *posterior* probability distribution $p(\text{ideal} \mid \text{observed})$, which describes the probability of being in an ideal state given the fact that we have made some observation. By making some ideal states more probable than others (given a fixed

---

[1] This will continue to be the case for the foreseeable future.

observation), we have regularized the problem of recovering the ideal state. Hence, such a probabilistic approach lets us impose prior knowledge, as was our goal. In this formulation, the solution to our problem, such as the denoised image or the estimated flow field, is recovered using probabilistic inference, for example by finding the ideal state that maximizes the posterior probability. Probabilistic models that directly model this posterior distribution are called *discriminative models*, as their goal is to discriminate between possible solutions given some observation. Alternatively, we can rewrite the posterior using Bayes' rule

$$p(\text{ideal} \,|\, \text{observed}) \propto p(\text{observed} \,|\, \text{ideal}) \cdot p(\text{ideal}), \tag{1.1}$$

which now consists of two components, the *likelihood* $p(\text{observed} \,|\, \text{ideal})$, and the *prior* $p(\text{ideal})$. The likelihood expresses a model of the observation process; that is it describes how the uncertain, measured data arises from the ideal state. Because of that, the likelihood is strongly application dependent; in image denoising, the likelihood describes the characteristics of the (sensor) noise that corrupted the ideal image. The prior models our a-priori belief of being in a particular ideal state without having yet made any observation. In case of image denoising it encodes our belief that the underlying ideal image is a particular image among all conceivable images. Probabilistic models that formulate the problem in this way and model each component separately are called *generative models*, because they model how the ideal state (and the observation) are generated[2]. One nice aspect of this approach is that it makes it directly apparent how Bayesian models can be used to impose prior knowledge on low-level vision problems. In this work we focus on generative approaches and model (and train) the prior and the likelihood separately.

From now on, we will refer to these prior distributions more succinctly as $p(\mathbf{x})$, where $\mathbf{x}$ is a vector that encodes the ideal gray levels, flow vectors, etc[3]. The observation model will be more concisely referred to as $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y}$ denotes the observed image or image sequence. But let us return to prior models in low-level vision and to the challenges that we are facing in devising them.

1.1 Challenges

Developing good prior models for low-level vision is a challenging problem for a variety of reasons. Note that we will mostly be discussing the case of modeling images here, but we should keep in mind that other types of low-level representations have very similar properties. The most obvious challenging property is that low-level vision representations

---

[2] Equivalently, generative models are sometimes thought of modeling the joint distribution $p(\text{ideal, observed})$.

[3] The notational conventions used in this work are summarized in Appendix A.

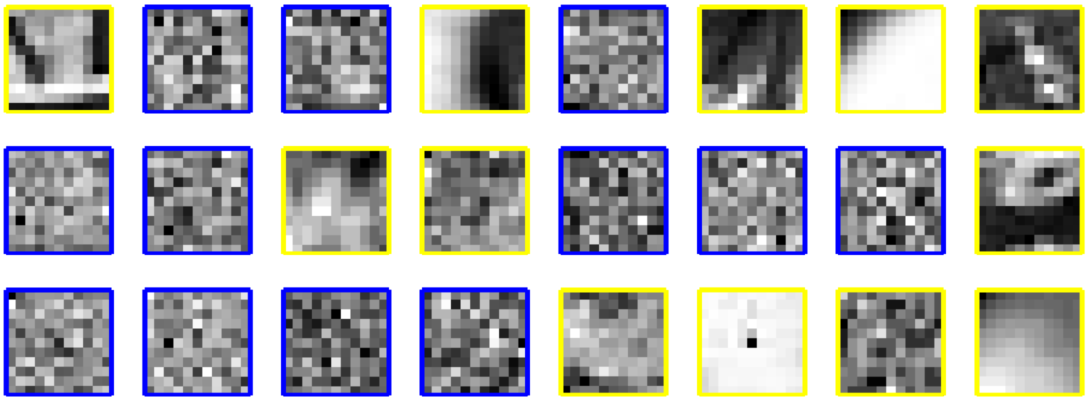


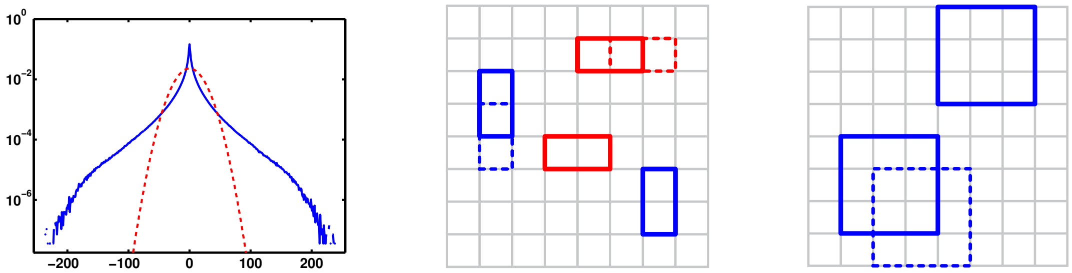
Figure 1.2: **Natural and random image patches.** The patches from natural images are framed in yellow, the random image patches are framed in dark blue.

are very high-dimensional. Even consumer-grade digital cameras take images with millions of pixels today, and even relatively low-quality imagery has hundreds of thousands of pixels. If we understand each pixel of the image as a component of a large random vector, then this random vector lives in a space of at least hundreds of thousands of dimensions. This alone may suggest that modeling such prior knowledge is challenging. If we for the moment suppose that we discretize these images with only two intensity values (black and white), then the binary vector representation of a moderate size image has at least 2^{100000} states. This means that we cannot really hope to enumerate all of the states or to use a trivial representation that "counts" how frequent each state is in naturally occurring images[4]. Despite the space being so huge, we as humans can relatively easily distinguish real images from unnatural ones. If we, for example, look at Figure 1.2, most of us can quite easily find the image patches in the figure that occur naturally versus those that are very unusual. Note that we can do that without even understanding what we see in these patches. This is an important observation, because it tells us that a high-level interpretation of an image is not necessary to classify it as natural or not. This implies that we should be able to model low-level prior knowledge without making a high-level interpretation explicit, although it is likely that high-level interpretations would help.

From this we can conclude that "real" images are quite special in the sense that they occupy only a tiny fraction of the space of all possible images; most images in that space will not look like real images at all (such as the blue examples in Figure 1.2). One important thing to note here, which actually further contributes to the difficulty of the problem, is that we do not wish to restrict ourselves to particular classes of images (or flow fields, depth maps, etc.). As opposed to modeling, for example, specific textures, our goal is to model

---

[4] We will define the notion of natural images more precisely in Section 4.1. For now it suffices to understand them as images that we typically encounter in real life.



(a) Marginal log-histogram of neighboring pixel differences and Gaussian fit.

(b) Modeling relations between pairs of neighboring pixels.

(c) Modeling relations over a larger neighborhood of pixels.

Figure 1.3: **Basic modeling principles for natural images.** (a) Non-Gaussian marginal statistics of natural images. The fit with a Gaussian distribution is shown with a dashed red line. (b) Modeling the spatial locality of natural images using simple pairwise neighborhood relations. The bright red pairs indicate relations between horizontally neighboring pixels; the dark blue ones between vertically neighboring pixels. The dashed pairs illustrate how relations are modeled on overlapping pairs. This setup is the basis of pairwise Markov random field models. (c) Modeling the spatial properties using larger neighborhoods (here 3×3 pixels). Here as well, relations are modeled on overlapping pixel regions; the dashed region shows one of the many overlapping regions. This setup corresponds to using high-order Markov random fields with large cliques.

the broad class of generic, but naturally occurring images, flow fields, and so on. In order to model priors over, say, natural images, we thus have to find a model that characterizes this small fraction of the space of all images. This turns out to be difficult, because this space has a number of complex properties. One important property revealed by the study of natural images [e. g., Ruderman, 1994] is the non-Gaussianity of natural images. If we look at simple marginal statistics of images, such as the statistics of two adjacent pixels in the image, we find that their distribution is strongly non-Gaussian. Particularly, if we look at the statistics of the intensity difference between neighboring pixels as shown in Figure 1.3(a), we find that there is a strong peak at zero (i. e., no intensity difference), but we also find so-called heavy tails. This means that there is substantial probability mass far away from zero, which indicates that large jumps in intensity occur with considerable frequency. One reason for these jumps are discontinuities in the image due to object boundaries. Heavy-tailed distributions are also called kurtotic, because the heavy-tailedness is characterized by large values of the kurtosis $\kappa = E\left[(x-\mu)^4\right] / E\left[(x-\mu)^2\right]^2$ of the distribution. We will defer a more detailed discussion of this and other related properties of natural images until Section 4.1, but knowing this already tells us that Gaussian models for low-level vision are likely to be inappropriate. This makes our task difficult, because non-Gaussian distributions are typically much harder to deal with in practice.

The other and for this work even more important property of natural images is their

(a) (b) (c)

Figure 1.4: **Typical pairwise MRF potentials and results.** (a) Example of a common robust potential function (negative log-probability). This truncated quadratic is often used to model piecewise smooth surfaces. (b) Image with Gaussian noise added. (c) Typical result of denoising using an ad-hoc pairwise MRF (obtained using the method of Felzenszwalb and Huttenlocher [2004]). Note the piecewise smooth nature of the restoration and how it lacks the textural detail of natural scenes.

spatial locality. By this we mean the fact that nearby pixels are often similar in their intensity, which we can easily convince ourselves of by looking at the natural examples (in yellow) in Figure 1.2. The reason for this is that close-by pixels are likely to come from the same object in the world, and many objects have a locally somewhat uniform appearance. This suggests that a good prior model of real images needs to model this locality. But what we loosely referred to as spatial locality is actually still quite complex. If we just model the fact that directly adjacent pixels have similar intensities and use such a model for image denoising, we obtain results that look unnatural. Figure 1.4 shows such an example, where major image discontinuities are preserved well, but most of the remaining real image structure has been smoothed away. Figure 1.3(b) illustrates how the pairwise neighborhood relations are structured in such a model; we should note that even though only some of the relations are drawn, they exist for all sets of overlapping neighboring pixel pairs. The reason for the relatively poor performance is that natural images and other low-level vision representations often exhibit rich spatial structures, resulting, for example, from complex surface textures or from complex 3D scene geometry. These structures often extend over a large neighborhood of pixels and can thus not be captured using simple relations between neighboring pixels. This strongly motivates that good prior models need to consider extended neighborhoods of pixels as is illustrated in Figure 1.3(c). This is the approach taken in this work, which we will make more precise in the following.

1.2 Approach

Let us first summarize the discussion so far: Formal models of image or scene structure play an important role in many vision problems where ambiguity, noise, or missing sensor data make the recovery of world or image structure difficult. In particular, models of *a* *priori* structure are used to resolve such problems by providing additional constraints that impose prior assumptions or knowledge about the kinds of structures that frequently occur. While prior knowledge can be supplied in many different ways, we will focus on probabilistic models of prior knowledge, which have a long history and provide a rigorous framework for treating uncertainty and combining different sources of information.

For problems in low-level vision such probabilistic prior models of the spatial structure of images or other scene properties are often formulated using graphical models, particularly as Markov random fields (MRFs) [Kashyap and Chellappa, 1981; Geman and Geman, 1984; Besag, 1986; Marroquin et al., 1987; Szeliski, 1990]. The important property of MRFs is that they directly allow us to model the local structure that exists in images and other spatial data. They do this by modeling each pixel of the image (or some other dense scene representation) with a node in a graph, and connecting the nodes based on some local neighborhood structure. The so-called maximal cliques of this graph (i. e., maximal subsets of nodes that are fully connected) give rise to factors in the underlying probability distribution, as we will see in more detail in the next chapter. The cliques thus impose a structure on the probability distribution of this MRF. Each clique has a so-called potential function associated with it, which indicates how compatible the nodes (i. e., pixels) in the clique are. In most cases MRFs have only modeled very simple, local structures using pairwise connections between neighboring pixels, in which case the maximal cliques are simply pairs of neighboring pixels. This setup is illustrated in Figure 1.3(b) in an informal fashion; more precise characterizations will be made in the subsequent chapter.

While these pairwise models are very widely used to this date [Sun et al., 2003; Tappen et al., 2003; Felzenszwalb and Huttenlocher, 2004], they have recently fallen short in terms of performance compared to quite different techniques that lack the generality and versatility of MRFs. The problem is not only that the small neighborhood systems with pairwise cliques limit the expressiveness, but also that many models have been hand-tuned and only crudely capture the statistics of natural images. A notable exception to this is the FRAME model by Zhu et al. [1998], which learns clique potentials for larger neighborhoods from training data by modeling the responses of a set of hand-defined linear filters.

On the other hand, prior models of small (image) patches have gained a lot of attention recently and have lead to very good results, also because they are typically learned from training data. Image patches have, for example, been modeled using a variety of sparse coding approaches or other sparse representations [Olshausen and Field, 1997; Hyvärinen et al., 2003; Teh et al., 2003]. These models do not just model relations between neighboring pixels, but model relations between all pixels in the patch, which is possible because the dimensionality of these patches is much lower than that of entire images. Such models have for example been applied to specific tasks such as image denoising [Welling et al., 2003], where they achieve much better results than can be obtained with simple pairwise MRFs. Many of these patch-based models, however, do not easily generalize to models for entire images, which has limited their impact on machine vision applications. Markov random fields, on the other hand, can be used to model the statistics of entire images, but their expressiveness has been restricted by the above limitations.

The main goal of this dissertation is to develop a framework for learning rich, generic prior models of low-level vision. The key insight is to exploit ideas from sparse image representations of image patches for learning Markov random field priors defined over large neighborhood systems, so-called *high-order Markov random fields*. The developed *Field of Experts* (FoE) models the prior probability of an image or another low-level scene representation in terms of a random field with large overlapping cliques, typically square patches of pixels. The potentials of each clique are represented as a Product of Experts [Hinton, 1999], a recently proposed method for learning high dimensional probability distributions. In this framework each potential is formed through a combination of a number of "experts". Each expert is a simple model that only describes a particular property of the neighborhood, but combining several of them leads to an expressive model. In particular, each expert models the response to a linear filter applied to the neighborhood of pixels. The key difference of the model over previous approaches such as the FRAME model [Zhu et al., 1998] is that it allows all parameters of the model including the linear filters to be learned from training data. Furthermore, in contrast to example-based approaches [e. g., Fitzgibbon et al., 2003], we develop a *parametric representation* that uses examples for training, but does not rely on examples as part of the representation. Such a parametric model has advantages over example-based models in that it generalizes better beyond the training data and allows for more elegant computational techniques. In contrast to patch-based models such as the Product of Experts [Teh et al., 2003], the FoE models entire images (or other dense scene representations) of an arbitrary size and is furthermore *translation-invariant*.

The increased flexibility of the proposed model comes at the expense of having many more model parameters that need to be chosen appropriately. Historically, the parameters of MRF models have often been chosen by hand [Geman and Geman, 1984; Geman and Reynolds, 1992], which, apart from not being very elegant, is infeasible for complex models with many parameters. Pseudo-likelihood methods [Besag, 1986] for estimating the model parameters from data are an exception to that, but have not found very widespread use. The goal here is to learn these parameters from data so that the model reflects the statistical properties of images or other low-level scene representations as well possible. The main reason for choosing parameters by hand in the past was that learning the parameters of a large MRF is computationally very difficult, because learning and probabilistic inference in many MRF models is NP-hard [Boykov et al., 2001]. To circumvent this problem, a large number of approximate learning and inference algorithms have been proposed in the recent literature [Boykov et al., 2001; Minka, 2001; Hinton, 2002; Yedidia et al., 2003; Wainwright et al., 2005]. Nevertheless, learning and inference in high-order MRF models with extended neighborhoods is actually particularly difficult. We address the task of learning the parameters of the proposed Fields-of-Experts model using an approximate learning algorithm known as contrastive divergence [Hinton, 2002]. While learning with this algorithm is still computationally intensive, it is efficient enough to be practical. The challenging problem of probabilistic inference will mostly be addressed using local optimization with gradient methods, but we will also discuss how more promising inference algorithms such as loopy belief propagation [Yedidia et al., 2003] could be applied to the FoE model.

The Field-of-Experts framework provides a principled way for learning MRFs from examples and the greatly improved modeling power makes them practical for complex tasks. We will use the FoE to model two kinds of scene representations, natural images and optical flow. To demonstrate the modeling power of the FoE model, we use it in three different applications: image denoising, image inpainting, and optical flow estimation. Image inpainting [Bertalmío et al., 2000] is the problem of filling in missing pixel values in an image without disturbing the overall image appearance. The need for image inpainting arises in situations where part of an image is "missing", for example because the photograph that the digital image was scanned from has scratches. Figure 1.5 illustrates the application of the FoE model for image denoising and image inpainting. Despite the generic nature of the prior and the simplicity of the approximate inference, we obtain state of the art results for these three applications that, until now, were not possible with MRF approaches.

1.3 Thesis Statement

The core thesis developed in this dissertation is that high-order Markov random fields, particularly the proposed Field-of-Experts framework, are more powerful models for representing prior knowledge in low-level vision than previous MRF models. Despite their complexity and the difficulty of learning in high-order MRF models in general, we demonstrate that learning can be made tractable using the method of contrastive divergence. Furthermore, we claim that these high-order models lead to competitive algorithms for varied low-level vision problems such as image denoising, image inpainting, and the estimation of image motion.

(a) (b)

(c) (d)

Figure 1.5: **Image restoration using a Field of Experts.** (a) Image from “Corel database” with additive Gaussian noise ($\sigma = 15$, PSNR = 24.63dB). (b) Image denoised using a Field of Experts (PSNR = 30.72dB). (c) Original photograph with scratches. (d) Image inpainting using the FoE model.

1.4 Motivation

Before introducing the technical details of Fields of Experts in Chapter 3, let us motivate the fundamental choices behind the model and discuss their current and future implications.

1.4.1 Advantages of probabilistic modeling

The proposed FoE model follows a probabilistic, and more specifically, a Bayesian approach. In our view, using probabilistic models for low-level vision applications has a number of important advantages over non-probabilistic methods, such as variational methods, which will be reviewed in Section 2.2.1. One advantage is that this lets us use the large literature on probabilistic modeling and inference with graphical models (reviewed in Section 2.1) for the purposes of low-level vision and gives us a powerful set of tools that is widely used, even outside of computer vision. One aspect of this (in general) is that we can use relatively efficient and well understood algorithms for inferring a denoised image, an optical flow field, etc. Such algorithms have been widely exploited in the low-level vision literature, including the belief propagation algorithm [Freeman et al., 2000; Tappen et al., 2003; Sun et al., 2003; Felzenszwalb and Huttenlocher, 2004] and graph cut techniques [Boykov et al., 2001; Kolmogorov and Zabih, 2002]. The advantage of these algorithms is that they are very good at finding approximate (and sometimes even exact) solutions of non-convex optimization problems [Tappen and Freeman, 2003; Szeliski et al., 2006]. Such non-convex problems occur in a variety of low-level vision applications, for example in optical flow estimation or stereo. Addressing them outside of the scope of graphical models often requires complex, special-purpose optimization techniques [e. g., Papenberg et al., 2006], whereas graphical models allow us to rely on standard tools[5]. Nonetheless, we will see in Section 3.4 that inference with the proposed high-order model is challenging due to its large cliques, even when studied within the well-understood framework of graphical models. The widespread use of graphical models, and Markov random field models in particular, will as we hope motivate many researchers to explore the problem of efficient learning and inference in high-order Markov random fields and create synergies between different domains.

The next, from the viewpoint of this work, even more important aspect of employing probabilistic models for low-level vision is that this allows us to actually choose suitable models based on their statistical properties. In many traditional approaches, it is very hard to know which kinds of prior models are suitable. Often, models are chosen based on simple considerations about what characterizes the data we are trying to model. Moreover, essentially all of these models have a number of parameters that need to be chosen appropriately.

---

[5] One could argue that special purpose optimization techniques may always perform better than standardized tools, but standard tools certainly help the proliferation of novel models.

In most of the cases these parameters are chosen in an ad-hoc, trial-and-error fashion. When viewed from a probabilistic point of view, we can assess our modeling choices in the context of statistical analyses of the data (such as the statistics of natural images, cf. Section 4.1). Furthermore, we can use learning algorithms to find suitable parameters [e. g., Zhu et al., 1998; Descombes et al., 1999; Krajsek and Mester, 2006], and in some cases we can even use statistical criteria, such as the likelihood of a test dataset to evaluate the goodness of our probabilistic model. Note that this does not necessarily invalidate many related approaches, because there are often intricate relations between probabilistic and non-probabilistic approaches. Instead, this is to promote a probabilistic view of the problem, which helps us choose a good model that is directly motivated by the statistics of our domain specific data.

Beyond this, probabilistic models have a number of other advantages, which in the context of future work may turn out to be quite important. First of all, probabilistic models give us an obvious way of assessing the uncertainly of any solution. In non-probabilistic approaches, such as variational methods, assessing uncertainty is difficult to do in general. Some authors have nevertheless proposed simple criteria for doing so [e. g., Bruhn et al., 2005], but these generally lack mathematical interpretability. In the probabilistic approach, we can compute (marginal) distributions over our solutions and, for example, use information theoretic measures, such as the entropy to assess uncertainty [MacKay, 2003]. Secondly, probabilistic approaches, specifically Bayesian ones, give us a principled way of combining information from several sources and permit us to maintain several solutions with their uncertainties. This way, we don't have to prematurely commit to a solution before we know whether the intermediate solution is sensible in the context of subsequent steps in a larger vision system (this is consistent with the principle of least commitment by Marr [1982, § 3.2]). Since low-level vision problems are often smaller pieces in a complete vision system, this is a very important advantage. In the recent years it has become a more widespread view that certain low-level vision problems cannot uniquely be solved without incorporating high-level knowledge, nor can more high-level tasks be solved without good models of low-level vision. This naturally suggests that low- and high-level knowledge (i. e., various layers of abstraction) have to be studied together and should be integrated in a joint framework. A number of authors have already proposed models along these lines [Tu et al., 2003; Kumar et al., 2005; Levin and Weiss, 2006; Rosenhahn et al., 2006]. In our view, the probabilistic framework is ideally suited to tackle such joint problems in a principled way. Hence, a contribution to better probabilistic models of low-level vision and a more thorough understanding of the modeling challenges there will hopefully facilitate building powerful models in the future that combine low- and high-level information.

1.4.2 Bayesian, but not "fully" Bayesian approach

Following a Bayesian approach is motivated by a number of factors, many of which have just been mentioned. The important ones are that this lets us take advantage of prior knowledge, and that Bayesian methods have been shown to be superior to "frequentist" approaches in many real applications [Jaynes, 2003]. Another important motivation not discussed so far is that Bayesian inference was shown to be a good model of human perception, for example of motion perception [Weiss and Adelson, 1998], and of cognitive judgements [Griffiths and Tenenbaum, 2006], which motivates its use in machine vision.

In contrast to what is sometimes assumed, Bayesian methods cannot just be characterized by making use of Bayes' rule and defining appropriate prior distributions. They can also be characterized by the fact that all intermediate quantities other than the ultimate solution that we are interested in are marginalized out. To make this distinction obvious, we refer to this as a "fully" Bayesian approach (Section 2.1.2 presents this in a slightly more formal fashion). An example of such quantities are the parameters of the model. Ultimately we are not really interested in parameters of the model, but instead we want to make predictions based on some measured data. In a fully Bayesian approach, such parameters are not determined ahead of time through learning, but instead they are integrated out (marginalized) during the inference process. This means that we can "average" models using their inherent uncertainty. This is particularly advantageous, if we are uncertain whether our model truly describes the process generating the data, and has proven effective in a number of applications [see e. g., Hoeting et al., 1999]. In practice, this is almost always the case, especially in computer vision where our current models are far away from modeling the full complexity of the physical world[6]. Nonetheless, fully Bayesian approaches are computationally demanding and can be difficult to apply in practice. Tipping and Bishop [2003] have applied this principle to image super-resolution; others have studied ways of making fully Bayesian treatments of large random field models more efficient in general [Qi et al., 2005; Welling and Parise, 2006]. In this dissertation we abandon a fully Bayesian treatment in order to limit the computational and mathematical complexity. Instead of marginalizing over the parameters of the model, we learn them from training data. But we note that in our view such a fully Bayesian treatment of vision problems seems highly desirable and a promising avenue for future work.

1.4.3 Discriminative versus generative approaches

As described above, there are two main approaches to probabilistic modeling, generative modeling and discriminative modeling. In discriminative modeling the posterior $p(\mathbf{x}|\mathbf{y})$ is

---

[6] This is another important argument for the use of probabilistic models in computer vision.

either modeled and trained directly, or if it is broken up into a likelihood and a prior, their parameters are trained jointly by maximizing the conditional likelihood [Lafferty et al., 2001]. This has the advantage that it focuses the capabilities of the model on the aspects that are most important with regards to the posterior distribution. If we ultimately only care about predictive inference with the posterior, then this initially seems like the most suitable approach. A number of models in the literature have followed this general approach [e. g., Lafferty et al., 2001; He et al., 2004; Ning et al., 2005; Kumar and Hebert, 2006]. Nonetheless, this procedure also has disadvantages. In particular, it requires that we train the entire model end-to-end for each possible application.

If we follow a generative approach, we independently train a model that describes how the underlying, true data $\mathbf{x}$ is generated (i. e., the prior), and a model that describes how the observed data $\mathbf{y}$ is generated from the true data (i. e., the likelihood)[7]. If we follow that approach, then we can use the very same generative prior model in a variety of applications only by changing the observation model. Since training is often expensive, this can amount to substantial computational savings. Because of that we follow the generative approach here, but note that the proposed Field of Experts could be used as component of a discriminative approach (see Section 3.6 for a more detailed discussion of this point).

1.4.4 Advantages of modeling entire images

The Fields-of-Experts model proposed here models the prior probability density of entire images, and other full, dense scene representations, which is an important aspect that deserves some further discussion. Alternatives to that are, as mentioned, models that only describe small patches (of images, flow, etc.). The general paradigm of modeling small patches has been successfully applied to a number of different problems, such as texture synthesis [Efros and Leung, 1999], or image denoising [Welling et al., 2003]. While these techniques may give very good results for certain applications, they may not be used as generally as the generic priors for entire images (or similar scene representations) that we develop here. If we have a generic model for the entire data, we can easily apply it to a range of different problems in a very similar fashion, as we will see in the later chapters. If we only have a model of small patches, then we need to work around the fact that we do not model the entire data and need to devise special purpose solutions for combining results from many patches into a global result. Moreover, in certain applications there may not even be any straightforward way of combining patches. Most importantly, even if such a model of patches is probabilistic, this does not immediately lead to a probabilistic

---

[7] Once again, in generative modeling we could instead train the joint distribution $p(\mathbf{x}, \mathbf{y})$, but this is rarely done in practice.

interpretation of entire images (or other dense scene representations).

1.5 Contributions and Overview

In Chapter 2 we review important basic concepts behind the model put forward here, including graphical models in general. We also review the related work on modeling prior distributions for low-level vision, which includes variational methods, previous MRF approaches, and patch-based prior models.

Chapter 3 introduces the Fields-of-Experts framework, a generic high-order Markov random field model for spatial data as it occurs in many low-level vision applications. The model combines advances in modeling data in small patches with MRF models that are based on spatially extended cliques. In particular, we show how Products of Experts can be used to model potentials of high-order Markov random fields. We go on to demonstrate how the parameters of the FoE model can be trained effectively using contrastive divergence. Training MRF models is a challenging problem in general, but is particularly hard for high-order MRF models. While the employed learning technique is generally not equivalent to maximum likelihood estimation, it allows us to train models much more quickly than using maximum likelihood, as the necessary Markov chain Monte Carlo sampler does not have to be run until convergence. In contrast to previous MRF models, the filters that are used to model the clique potentials are not chosen by hand, but instead learned from training data during this training phase. We also discuss the issue of probabilistic inference with FoE models, compare the framework to related models, and discuss a number of other details in order to improve the understanding of the model.

In Chapter 4 we first review the literature on natural image statistics. To study the application of Fields of Experts to modeling natural images, we train the model on a standard database of natural images [Martin et al., 2001]. We show how it can be applied in a rather straightforward fashion to the problems of image denoising and image inpainting. The algorithm used for inferring the denoised or inpainted image is shown to have interesting connections to nonlinear diffusion techniques [Weickert, 2001], but in case of the FoE model many more linear filters are used. We go on to demonstrate that the FoE outperforms simple, pairwise MRF models and show that despite its generality and its wide applicability, it is competitive with the state-of-the-art in both image denoising and image inpainting. It is particularly important to note that such results had previously not been reported with other Markov random field approaches. We furthermore present a detailed evaluation of the impact of a number of modeling aspects of the FoE model on application performance.

Chapter 5 illustrates the use of the FoE model in the area of image motion (optical flow) modeling and estimation. This application is challenging and especially interesting, because as opposed to natural images where training data is readily available in form of digital images, ground truth optical flow cannot directly be measured. We develop a novel database of “natural” optical flow fields by synthesizing them from realistic 3D geometry data and a realistic database of camera motions. Based on this database, we provide a detailed analysis of the statistical properties of optical flow. We show how Fields of Experts can be used to model the spatial statistics of optical flow and propose an optical flow algorithm based on this model, which is shown to outperform previous, related optical flow techniques.

In Chapter 6 we summarize the findings in this dissertation and describe the limitations of the approach. We also discuss directions that seem to be worthwhile to follow in the future.

CHAPTER 2

Background and Related Work

In this chapter we will review the basic concepts of graphical models, as well as the related work on modeling prior knowledge in low-level vision. A number of other topics and methods will be reviewed in the later chapters, since their knowledge is not crucial to understanding the core part of this dissertation. This includes the statistics of natural images (Section 4.1), image denoising (Section 4.3.1), and optical flow estimation (Section 5.1.1).

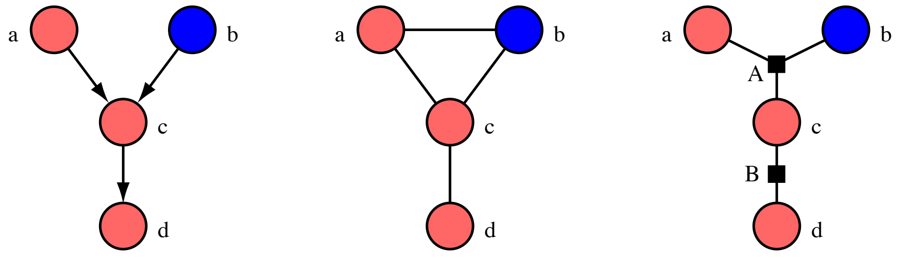
2.1 Graphical Models, Learning, and Inference

In order to fully understand the Field-of-Experts model put forward in this dissertation, it is necessary to understand the important basic aspects of graphical models in general, as well as of learning and inference in graphical models. Of course, a thorough review is well beyond the scope of this work. Hence, we will only review the aspects that are most relevant and refer the reader to more thorough discussions [Lauritzen, 1996; Wainwright and Jordan, 2003; Bishop, 2006].

Graphical models are a widely used tool for both understanding and formalizing probability distributions[1] over many variables. They have been successfully applied to a wide variety of problems from diverse domains, such as natural language, computational biology, robot navigation, and of course computer vision, particularly also low-level vision. Graphical models furthermore provide a link into the graph theoretical foundations of algorithmic theory in computer science. In particular, the graph structure tells us how the corresponding probability distributions factorize; that is how they break into simpler pieces (both graphically and mathematically). We denote the d-dimensional random vector that we would like to model as $\mathbf{x}$. A graphical model is a graph $G = (V, E)$ that associates a

---

[1] We use this terminology loosely, as we will not strictly distinguish between discrete and continuous random variables here.



(a) Directed graphical model.

(b) Undirected graphical model / Markov random field.

(c) Factor graph.

Figure 2.1: **Three kinds of graphical models.** The dark blue node denotes an observed random variable; all other random variables are hidden.

node $v \in V$ with each random variable[2] of $\mathbf{x}$, which we denote as x_v. Nodes or random variables can either be observed, because their value is known to us ahead of time (i. e., the graphical model defines a conditional probability given these observed nodes), or they can be hidden and need to be inferred. The edges $e \in E$ of the graph can be either directed or undirected, and (as usual in graphs) connect pairs of nodes.

Directed graphical models. In a directed graphical model (also known as Bayesian network) [Lauritzen, 1996], all edges e are directed and form a directed acyclic graph (i. e., a graph with no directed cycles). Figure 2.1(a) shows a small example graph. In order to understand the factorization structure, we define the set of parents $\pi(v)$ as the set of all nodes that have an edge going *into* v (this set is empty if there are no parents). We then write the probability distribution under such a directed model as

$$p(\mathbf{x}) = \prod_{v \in V} p(x_v | \mathbf{x}_{\pi(v)}), \tag{2.1}$$

where $\mathbf{x}_{\pi(v)}$ are all the components of the random vector that belong to the parents of v. This means that a directed graphical model tells us how a probability distribution factors into conditional distributions over subsets of nodes. If some nodes are hidden (e. g., $\mathbf{x}_H$), but others observed ($\mathbf{x}_O$), then the graphical model defines the conditional probability $p(\mathbf{x}_H | \mathbf{x}_O)$ using essentially the same rule as above (observed nodes are omitted from the product). In the example in Figure 2.1(a), the probability distribution under the model factorizes as

$$p(a, c, d | b) = p(d|c) \cdot p(c|a, b) \cdot p(a). \tag{2.2}$$

---

[2] In abuse of notation, we do not formally distinguish between random variables and a particular value that a random variable can assume.

Directed graphical models have very nice properties, for example, that their underlying probability distributions are given in terms of conditional probability distributions, and that they identify the causal structure behind the model. But the restriction that the graph needs be acyclic is too strong for many vision applications, and certainly for the majority of low-level vision applications.

Undirected graphical models. Undirected graphical models do not possess this restriction; the edges are undirected and the graph may have arbitrary cycles. In an undirected graphical model, the factorization structure depends on the cliques C of the graph; that is the subsets of nodes that are fully connected. Associated with each clique $c \in C$ is a potential function $f_c : V_c \to \mathbb{R}_+$, which assigns a positive compatibility score to the random variables $\mathbf{x}_{(c)}$ of the clique. The joint probability distribution factors as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C} f_c(\mathbf{x}_{(c)}), \tag{2.3}$$

and is sometimes equivalently written in terms of a Gibbs distribution (or Gibbs random field):

$$p(\mathbf{x}) = \frac{1}{Z(T)} \exp\left\{-\frac{1}{T} E(\mathbf{x})\right\} = \frac{1}{Z(T)} \exp\left\{-\frac{1}{T} \sum_{c \in C} U_c(\mathbf{x}_{(c)})\right\}. \tag{2.4}$$

In abuse of nomenclature, we also refer to the $U_c : V_c \to \mathbb{R}$ as potential functions as they are functionally equivalent to the potentials from Eq. (2.3). The function $E(\mathbf{x}) = \sum_c U_c(\mathbf{x}_{(c)})$ is the energy of the system; T is a "temperature" that is often assumed to be 1. The term

$$Z(T = 1) = Z = \int_{\mathbf{x}} \prod_{c \in C} f_c(\mathbf{x}_{(c)})\, d\mathbf{x} = \int_{\mathbf{x}} \exp\left\{-\sum_{c \in C} U_c(\mathbf{x}_{(c)})\right\} d\mathbf{x} \tag{2.5}$$

is a normalization constant that ensures that $p(\mathbf{x})$ integrates (or in the discrete case sums) to 1. In the context of Gibbs distributions this term is also known as the partition function [e. g., Li, 2001, §1.2.3], especially in the statistical physics literature. Note that the potential functions f_c themselves do not have to be normalized. If we have hidden variables $\mathbf{x}_H$, the partition function is dependent on them, i. e., $Z(\mathbf{x}_H)$.

While this formalism is more general and allows for cycles, it also has disadvantages, such as that in general undirected graphical models there is no direct connection between the potential functions and marginal distributions of subsets of nodes[3]. Also, computing the normalization constant Z is typically intractable. For example, in discrete models the space that needs to be summed over is exponentially large, and brute force summation

---

[3] Section 4.2 discusses special cases where there are in fact such direct connections.

can often not be avoided. Both these aspects make learning and inference in undirected graphical models hard.

Figure 2.1(b) shows an example of an undirected graphical model, which is in fact a generalization of the directed model in Figure 2.1(a). This simple graph already illustrates one of the drawbacks of the undirected formalism. Because the cliques are not unique (smaller cliques may be the subset of larger cliques), we can write the probability distribution for this example graph either as $p(a, c, d|b) = \frac{1}{Z(b)} f_1(c, d) \cdot f_2(a, b, c)$ or as $p(a, c, d|b) = \frac{1}{Z(b)} f_1(c, d) \cdot f_2(a, b) \cdot f_3(b, c) \cdot f_4(a, c)$. To remedy that, it is sometimes assumed that the cliques be maximal, which makes the factorization unique. Another possibility is to use a different formalism, such as factor graphs discussed below.

Markov random fields. Markov random fields (MRFs) [Li, 2001] consist of a random vector $\mathbf{x}$ (equivalent to a set of random variables, which we call V as above), as well as a neighborhood structure $\mathcal{N}$ on these variables, and fulfill the following two requirements: (1) Positivity, i. e., $p(\mathbf{x}) > 0$; (2) Markovianity, i. e., $p(x_v|\mathbf{x}_{V\setminus\{v\}}) = p(x_v|\mathbf{x}_{\mathcal{N}(v)})$. Here, $V \setminus \{v\}$ is the set of all variables except v itself, and $\mathcal{N}(v)$ is the set of all neighbors of v. The Markov property says that given the set of neighbors $\mathcal{N}(v)$ the variable v is conditionally independent of all other variables in the random field. Because of that, the set $\mathcal{N}(v)$ is also called the Markov blanket of v. The most important theoretical result is the Hammersley-Clifford theorem [Moussouris, 1974], which states that Markov random fields with a neighborhood system $\mathcal{N}$ are equivalent to Gibbs random fields where the edges E obey the same neighborhood system (i. e., $(i, j) \in E \Leftrightarrow i \in \mathcal{N}(j)$). Because of that, Markov random fields and undirected graphical models describe the same class of probability distributions; we henceforth use this terminology interchangeably. Nevertheless, it is important to note that the potentials of a Markov random field do not (in general) have simple closed-form relations to the conditional distributions that govern the Markov property.

Historically, Markov random fields have often been understood to mean special kinds of undirected graphical models, mainly those that are arranged on a lattice. Today, it is customary to not make this distinction anymore; we will use either terminology in this work.

Factor graphs. Factor graphs [Kschischang et al., 2001] remove the formal ambiguity of undirected graphical models by explicitly formalizing the factorial structure. In addition to the variable nodes, there is another set of nodes $F \in \mathcal{F}$, the so-called factor nodes, which directly correspond to factors in the factorized distribution. The graph forms a bipartite graph, where each edge connects a variable node to a factor node. The probability distribution under a factor graph is written as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{F \in \mathcal{F}} f_F(\mathbf{x}_{(F)}), \tag{2.6}$$

where $\mathbf{x}_{(F)}$ are all the variable nodes connected to factor F. Factor graphs are a generalization of both directed and undirected graphical models. Figure 2.1(c) shows an example graph that makes one of the possible factorization structures of the MRF in Figure 2.1(b) obvious. In the example, the probability distribution is written as $p(a, c, d|b) = \frac{1}{Z(b)} f_B(c, d) \cdot f_A(a, b, c)$.

We distinguish between two different types of factor graphs (or undirected graphical models): (1) Those with factors (or maximal cliques) that connect only pairs of nodes. We will call these *pairwise Markov random fields.* (2) Those with factors (or maximal cliques) that connect more than two nodes. These are the so-called *high-order Markov random fields*[4]. Especially for the high-order MRFs used in this dissertation, the factor graph formalism makes is easier to understand the model, which is why it is adopted throughout the work. Despite that, we usually use the terms factors and maximal cliques interchangeably.

2.1.1 Inference

Inference in graphical models is the task of computing information about the (hidden) variables $\mathbf{x}_H$, given the observed variables $\mathbf{x}_O$. Typically, this means one of two things: (1) We want to compute a configuration $\mathbf{x}_H^*$ that maximizes the posterior probability $p(\mathbf{x}_H|\mathbf{x}_O)$. This is called *maximum a-posteriori* (MAP) estimation. (2) We want to compute the *marginal distribution* $p(x_i|\mathbf{x}_O)$ over a single hidden node i, or marginal distributions over sets of hidden nodes: $p(\mathbf{x}_{(s)}|\mathbf{x}_O)$. Here s denotes a set of hidden nodes in the graph; in most of the cases s is a clique of the graph. Another important inference problem is that of computing the partition function Z of an undirected graphical model.

In general, exact inference in graphical models, especially those with cycles, is very difficult [Jordan and Weiss, 2002]. For example, MAP estimation in discrete graphical models is NP hard in general [Shimony, 1994]. Even more, constant ratio approximations were shown to be NP hard as well [Abdelbara and Hedetniemib, 1998]. Nevertheless, there are polynomial time algorithms for certain classes of graphical models. Inference is efficient for tree-structured models using Pearl's belief propagation (BP) algorithm [Pearl, 1988; Lauritzen, 1996], which is linear in the number of nodes. On the other hand, inference is considerably more difficult for loopy graphs (i. e., those with cycles). But even in that domain, there

---

[4] While it is, in principle, possible to convert high-order models into pairwise models [Yedidia et al., 2003], doing so is often very inefficient in practice, because this significantly increases the size of the state space.

are polynomial time algorithms for certain classes of MRFs [Kolmogorov and Zabih, 2004; Kolmogorov and Rother, 2007]. Also, it has been shown that even for problems where no polynomial time algorithm is known, exact MAP inference can done in practice even for very large graphs [Meltzer et al., 2005; Yanover et al., 2006]. In many practical cases, however, there are no known exact inference algorithms that are computationally feasible. Hence, for many applications of graphical models we have to settle for *approximate inference*, which we will review in the following.

Approximate inference algorithms. Historically, approximate inference with undirected graphical models has usually relied on sampling, particularly on *Markov chain Monte Carlo (MCMC)* sampling (see [Neal, 1993], [Andrieu et al., 2003], and [Gelman et al., 2004, § 11] for overviews). The idea is that if we can sample the posterior distribution $p(\mathbf{x}_H|\mathbf{x}_O)$ over the hidden variables, then we can approximate marginal distributions using the samples. Since the distributions underlying most graphical models cannot directly be sampled, sampling usually relies on the Markov chain Monte Carlo method [Neal, 1993], where a Markov chain is set up so that its stationary distribution is the posterior distribution of interest. Many different types of MCMC samplers exist [see e. g., Andrieu et al., 2003]; the Gibbs sampler [Geman and Geman, 1984] is one of the most popular ones. We will briefly review several other MCMC samplers in the context of the proposed FoE model in Section 3.3.3.

We can also do approximate MAP estimation using MCMC. One simple way is to draw many samples and to report the one with the largest posterior probability. Simulated annealing [Geman and Geman, 1984] is a more sophisticated variant, which is particularly interesting due to its global convergence properties. Nevertheless, for many types of graphical models sampling-based inference is computationally very expensive.

Another large class of approximate inference algorithms are those based on *variational methods* [see e. g., Jordan et al., 1999]. Instead of doing inference on the original model, inference is performed on a restricted family of models where inference is easy, but is done for the member of that restricted family that most closely resembles our posterior distribution of interest. The mean field approximation [e. g., MacKay, 2003, § 33] is a popular such approach. More recently, another family of variational algorithms has become even more popular: Pearl's belief propagation algorithm [Pearl, 1988] was applied even to graphs with loops, where it no longer gives exact marginals or MAP solutions, but still shows impressive performance [Weiss, 1997]. This message-passing algorithm can be applied to arbitrary factor graphs [Kschischang et al., 2001]. BP was later related to (locally) minimizing the so-called Bethe free energy, an important class of approximations in statistical physics [Yedidia et al., 2003, 2005], which helped explain its good performance. A large number of extensions and related methods have been developed over the years including stronger variational approximations [Yedidia et al., 2005], convergent extensions [Yuille, 2002], different approximation measures [Minka, 2005], and methods providing provable bounds [Wainwright and Jordan, 2003].

A final class of inference techniques solely related to MAP estimation is based on *(local) optimization*. Iterated conditional modes (ICM) [Besag, 1986] is a classical technique in this area; deterministic annealing [Hofmann et al., 1998] is another one. More recently and in particular for discrete models, another family of methods based on graph cuts has become popular [Boykov et al., 2001]. The idea is that for certain types of Markov random fields, good MAP approximations can be obtained by solving min-cut/maximum-flow graph problems. More recently, it has been shown that such techniques can in fact even give exact MAP solutions for certain types of models [Kolmogorov and Zabih, 2004; Kolmogorov and Rother, 2007]. While marginal distributions cannot be obtained using graph cut methods, it is possible to obtain max-marginals [Kohli and Torr, 2006]. Other approaches are based on mathematical programming relaxations to MAP estimation [Yanover et al., 2006; Kumar et al., 2006].

As will be discussed in Section 3.4, a number of these inference methods have practical limitations when applied to MRF models of high order. While inference is generally difficult for loopy graphs, it is particularly difficult (in a computational sense) for loopy graphs of high order.

2.1.2 Learning

So far, we have described undirected graphical models in an abstract form, where the potential functions are arbitrary positive functions. But to make these models describe a particular probability distribution, we have to specify suitable potential functions. Usually, the potentials are based on some parametric family of functions, which allows the model to describe a range of different distributions. If we adopt a Bayesian approach, the parameters of the potential functions are treated as random variables themselves, which means that they can be regarded as hidden nodes in the graph. If we separate the parameters Θ from the other hidden variables, we can write the posterior as $p(\mathbf{x}_H, \Theta|\mathbf{x}_O)$. During inference where we are only interested in $\mathbf{x}_H$, these parameter nodes are then marginalized out:

$$p(\mathbf{x}_H|\mathbf{x}_O) = \int p(\mathbf{x}_H, \Theta|\mathbf{x}_O)\, d\Theta. \tag{2.7}$$

For many practical problems, especially those of interest here, such a fully Bayesian treatment is currently computationally infeasible. Instead, we can determine a suitable set of parameters Θ^* ahead of time using learning, and then use these learned parameters during inference[5]:

$$p(\mathbf{x}_H|\mathbf{x}_O) \approx p(\mathbf{x}_H|\mathbf{x}_O;\,\Theta^*). \tag{2.8}$$

There are a large number of different ways of learning parameters of graphical models, hence we will only review the most important ones. The models used in this dissertation are always trained in a fully observed setting; that is *during learning* we have training data available that describes both the observed variables $\mathbf{x}_O$ as well as the normally hidden variables $\mathbf{x}_H$. The case of learning with unobserved (i. e., missing) data is not discussed here. The most popular criteria for learning in graphical models are the *maximum likelihood (ML)* approach [Geyer, 1991][6], and the *maximum a-posteriori (MAP)* approach. In ML estimation, we determine the model parameters Θ^* by maximizing the likelihood $p(\mathcal{X};\,\Theta)$ of the training dataset $\mathcal{X}$. In MAP estimation, we additionally impose a prior $p(\Theta)$ on the parameters. Both ML and MAP estimation are often difficult problems, because the normalization term that is implicit in Eq. (2.8) depends on the parameters Θ. This means that in order to estimate Θ using ML or MAP, we have to be able to estimate $Z(\Theta)$ (or at least compute how $Z(\Theta)$ depends on Θ). But estimating the normalization term is one of the difficult inference problems that we discussed above. Since learning relies on probabilistic inference, both ML and MAP learning are computationally expensive. This is described in more detail in Section 3.3.1 in conjunction with the concrete model introduced in this dissertation.

A number of alternative learning criteria have been proposed over the years that alleviate the problem of having to estimate the normalization term. They include maximum pseudo-likelihood [Besag, 1986], score matching [Hyvaärinen, 2005], and discriminative training of energy-based models [LeCun and Huang, 2005]. Different algorithms for maximum likelihood estimation, such as generalized iterative scaling [Darroch and Ratcliff, 1972; Pietra et al., 1997], have been used as well. Another efficient learning rule called contrastive divergence [Hinton, 2002], which is closely related to ML estimation, will be described in more detail in Section 3.3.2.

2.2 Modeling Prior Knowledge in Low-Level Vision

After introducing the necessary background on graphical models, we will now review the related work on modeling prior knowledge. The need for modeling prior knowledge in solving problems in low-level vision has long been recognized [Geman and Geman, 1984;

---

[5] Note that this does not imply that we are necessarily abolishing the probabilistic treatment of parameters. We may, for example, still impose priors on the parameters.

[6] See also [Silvey, 1975, § 4] for a more general overview and the asymptotic properties of ML estimation.

Poggio et al., 1985]. Many problems in low-level vision are mathematically *ill-posed* [Hadamard, 1923], often because there is no unique solution to the problem without making additional assumptions. One obviously ill-posed problem that illustrates this is image super-resolution [e. g., Tappen et al., 2003], where the goal is to find an image that has a higher spatial resolution than the input image. Because there are more variables to be solved for than there are input variables, it is necessary to make assumptions that constrain the space of admissible solutions. Similar challenges exist, for example, in optical flow estimation [Horn and Schunck, 1981], as we discussed in the previous chapter. As mentioned, it is necessary to make prior assumptions about the space of solutions. Constraining the space of solutions in order to solve ill-posed problems is traditionally referred to as regularization and was formally introduced by Tikhonov [1963].

There are various ways of thinking about regularization and modeling prior knowledge in low-level vision. One approach is to view low-level vision problems in a deterministic way, and to impose deterministic prior knowledge. The variational approach [e. g., Poggio et al., 1985] is a popular framework for deterministic regularization, which will be reviewed below.

The other predominant approach is to view the problem in a probabilistic sense. A probability distribution describes our a-priori assumptions about the solution space, in that certain solutions are viewed as being more probable than others. This is the approach followed in this work, which was initially made popular through the introduction of Markov random field models into low-level vision [Wong, 1968]. Further below we will review the most prominent MRF models of prior knowledge that have been developed for low-level vision applications. Moreover, we will also review other probabilistic models including sparse coding methods.

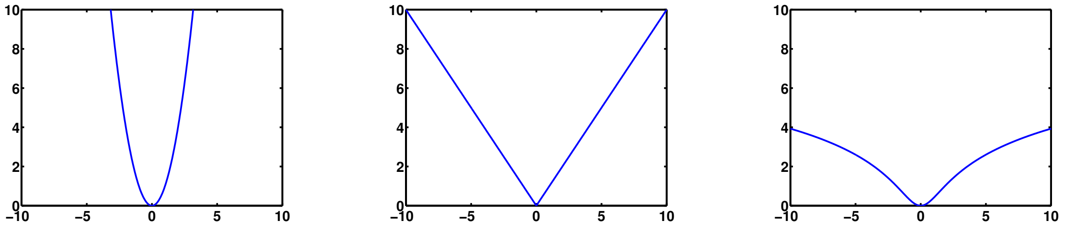
Interestingly, there exist direct connections between deterministic and probabilistic regularization that have been known for some time [Marroquin et al., 1987]; some of these connections will be discussed in the following as well.

2.2.1 Variational approaches and related regularization methods

Many traditional methods of imposing prior knowledge for low-level vision problems are based on variational formulations [Terzopoulos, 1986; Marroquin et al., 1987; Schnörr et al., 1996][7]. There the sought after low-level representation is regarded as a function $f(x, y)$ of real-valued coordinates (e. g., $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$). A general formulation of this framework is given by Marroquin et al. [1987]: The goal is to find $f(x, y)$ given some observation $o(x, y)$

---

[7] Note that this is not to be confused with variational inference methods for probabilistic models.



(a) Quadratic penalty $\rho(y) = y^2$.

(b) Total variation penalty $\rho(y) = |y|$.

(c) Lorentzian robust penalty $\rho(y) = \log(1 + \frac{1}{2}y^2)$.

Figure 2.2: **Penalty functions for regularization.** The quadratic penalty in (a) strongly penalizes large deviations. The TV penalty in (b) and the Lorentzian penalty function in (c) penalize large deviations less, which allows for discontinuities in the regularized data.

by minimizing an energy functional of the type

$$E(f;\, o) = \iint [(A \circ f)(x,y) - o(x,y)]^2 + \lambda \cdot [(P \circ f)(x,y)]^2 \, dx \, dy, \tag{2.9}$$

where $A \circ f$ and $P \circ f$ are the applications of application dependent (linear) operators to f. The first term, called the data-term, ensures that the recovered structure f (e. g., an image) is sensible given the observation o, and the second term, called spatial term, regularizes the problem (see above). In the concrete application of image restoration (denoising), such a functional is given by:

$$E(f;\, o) = \iint \left(f(x,y) - o(x,y)\right)^2 + \lambda \cdot \left(f_x(x,y)^2 + f_y(x,y)^2\right) \, dx \, dy, \tag{2.10}$$

where $f_x(x,y)$ is the x-derivative of the low-level scene representation at point (x,y), $f_y(x,y)$ is the y-derivative, and λ is a weight that allows adjusting the relative influence of the spatial versus the data term. The data term here ensures that the restored (i. e., denoised) image is close to the measured (i. e., noisy) observation; the spatial term penalizes large image gradients, and thus ensures that the restored image varies smoothly. This general formulation found widespread use throughout low-level vision, for example in image restoration [see e. g., Schnörr, 1994], and optical flow estimation [Horn and Schunck, 1981].

The problem with these particular formulations is that the solutions are typically too smooth, in the sense that the restored image or the recovered optical flow field does not have discontinuities as they naturally exist in real data. To overcome this, various solutions have been proposed. One is to explicitly model the discontinuities and prohibit any smoothing across the discontinuities. This is the approach followed in the classical variational framework by Mumford and Shah [1989]. Another possibility is to regularize using a different penalty function [Terzopoulos, 1986; Blake and Zisserman, 1987]. We can rewrite the above variational formulation in terms of a general penalty function $\rho(\cdot)$, e. g.:

$$E(f;\, o) = \iint (f(x,y) - o(x,y))^2 + \lambda \cdot [\rho(f_x(x,y)) + \rho(f_y(x,y))] \; dx\, dy. \tag{2.11}$$

One such example is total-variation (TV) regularization [cf. Rudin et al., 1992], where $\rho(y) = |y|$ (see Fig. 2.2(b)). The TV norm does not penalize large gradient values as much as the quadratic penalty $\rho(y) = y^2$ used above (see also Fig. 2.2(a)). Because of that, it allows occasional discontinuities and leads to results of better quality. Other penalty functions used for discontinuity-preserving regularization are motivated by robust statistics [Geman and Reynolds, 1992; Black and Anandan, 1996; Black et al., 1998]. Many of these discontinuity-preserving regularization methods use non-convex penalty terms, such as the Lorentzian robust penalty shown in Figure 2.2(c). While such non-convex regularizers often lead to superior results, they make finding a minimizer of these energy functionals very difficult.

Variational approaches of various forms have continued to be popular and quite successful, for example in optical flow estimation [Bruhn et al., 2005], motion segmentation [Cremers and Soatto, 2005], multi-view stereo [Jin et al., 2002], and image restoration [Bertalmío et al., 2000].

Variational methods bear connections to a number of other regularization methods. In particular, they are directly connected to partial differential equation (PDE) methods such as nonlinear diffusion [Weickert, 2001], which in many cases can be derived as the Euler-Lagrange equations of an energy functional. In Section 4.3.1 we will review a number of such methods and their application to the problem of image denoising.

Another connection, which is more important from the viewpoint taken in this dissertation, is that to probabilistic methods. In particular, if variational energy functionals are discretized in space (i. e., so that the coordinates x and y can only be integers), then they can be interpreted in a probabilistic sense [Mumford, 1994], particularly as Markov random fields [Marroquin et al., 1987; Szeliski, 1990]. Suppose we are trying to discretize the image restoration functional from Eq. (2.10). As first approximation, image gradients can be computed on a discrete grid as the pixel differences between horizontally and vertically neighboring pixels. A discrete version of Eq. (2.10) is thus given by the following energy function:

$$E(\mathbf{X};\, \mathbf{O}) = \sum_{i=1}^{I} \sum_{j=1}^{J} (X_{i,j} - O_{i,j})^2 + \lambda \cdot \left((X_{i+1,j} - X_{i,j})^2 + (X_{i,j+1} - X_{i,j})^2 \right). \tag{2.12}$$

Here, $\mathbf{X}$ is a 2D-array of all pixels of the sought after data (e. g., an image), $\mathbf{O}$ is an array of pixels from the measured observation, and i and j range over all pixels in the grid. But this

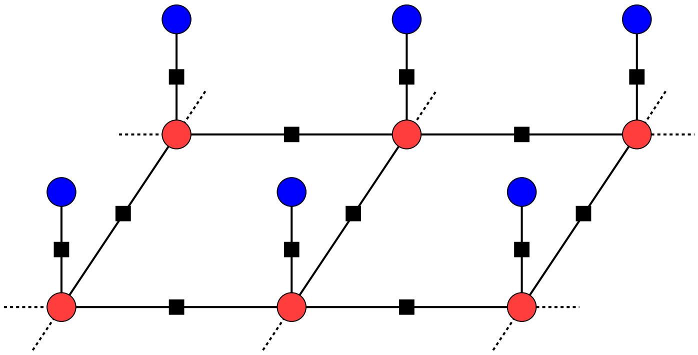


Figure 2.3: **Pairwise MRF model for image denoising.** The dark blue nodes denote the observed, noisy pixels. The light red nodes denote the pixels of the denoised image that need to be inferred.

is exactly the energy (negative log probability) of a Gibbs random field with the probability distribution

$$p(\mathbf{X}|\mathbf{O}) = \frac{1}{Z} \prod_{i=1}^{I} \prod_{j=1}^{J} \exp\left\{ - (X_{i,j} - O_{i,j})^2 - \lambda \cdot \left((X_{i+1,j} - X_{i,j})^2 + (X_{i,j+1} - X_{i,j})^2 \right) \right\}. \tag{2.13}$$

Figure 2.3 shows the graphical model structure of such a pairwise Markov random field for image restoration (denoising). Other discretizations are possible as well and lead to slightly different MRF models [Szeliski, 1990]. Beyond what we discussed so far, there are also notable connections between robust regularization methods as mentioned above and MRF models based on so-called line-processes [Black and Rangarajan, 1996].

While we pursue a probabilistic framework in this work, many of the developed methods are, due to this connection, applicable in the context of deterministic regularization as well. This suggests that the FoE framework developed in Chapter 3 could be combined with a varied class of variational approaches.

2.2.2 Pairwise Markov random field models

Many problems in low-level vision are formulated using prior knowledge imposed through probabilistic models, particularly using Markov random fields [Marroquin et al., 1987; Szeliski, 1990] (see also [Pérez, 1998; Li, 2001] for recent overviews). MRF models for spatial data have been developed as early as the 1960s, for example by Abend et al. [1965] and Wong [1968]. They were slowly popularized later [Woods, 1972], but only started to have widespread use in low-level vision in the 1980s, particularly in image restoration [Kashyap and Chellappa, 1981; Geman and Geman, 1984; Besag, 1986]. Other applications include optical flow estimation [Murray and Buxton, 1987; Konrad and Dubois, 1988], and texture modeling [Cross and Jain, 1983; Gimel'farb, 1996; Zhu et al., 1997]. In almost all of the mentioned cases, the neighborhood structure of the MRF is chosen by hand, although the type of edge structure and the choice of potentials varies substantially.

In such a formulation the nodes of the model are arranged on a 2D lattice that corresponds to the spatial organization of the image, flow field, depth maps, etc. Each node corresponds to pixel intensities, or image-like data such as range values, surface normals, or optical flow vectors; we usually refer to all of these as "pixels". Each pixel of the observation $\mathbf{y}$ constitutes an observed node in the graph; each pixel of the sought after data $\mathbf{x}$ is a hidden node in the graphical model. As we discussed in Chapter 1, Bayesian approaches to low-level vision have two components, an observation model or likelihood $p(\mathbf{y}|\mathbf{x})$, and a prior $p(\mathbf{x})$. The observation model is embodied in the graphical model through edges between the observed nodes and the hidden nodes. This likelihood model is application dependent as mentioned above, and will be discussed in more detail in Chapters 4 and 5. For sake of concreteness, let us consider the application of image denoising. There, one typically assumes that the noise at each pixel is independent of the other pixels, that the noise is additive, and that the noise characteristics are the same for all pixels. Hence, we can formulate such a likelihood with an edge between each observed pixel and the corresponding hidden pixel:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_L} \prod_i f_L(y_i - x_i). \tag{2.14}$$

f_L is a potential (factor) that models the observation noise at each pixel.

In the simplest sensible way, the prior $p(\mathbf{x})$ is formed by connecting each hidden node in the graph to its 4 direct neighbors (left, right, top, and bottom) and forms a pairwise MRF [Geman and Geman, 1984; Besag, 1986; Marroquin et al., 1987; Szeliski, 1990; Tappen et al., 2003]. While being simple, this nevertheless poses an important restriction on the kinds of structures in an image, depth map, flow field, etc. that can be modeled. It is also usually assumed that the MRF model is homogeneous; that is the potentials do not depend on the spatial location within the image grid. This property gives rise to the translation-invariance of an MRF model[8]. Such a pairwise MRF prior is thus written as

$$p(\mathbf{x}) = \frac{1}{Z_{\mathrm{pw}}} \prod_i \prod_{\substack{j \in \mathcal{N}(i), \\ j > i}} f_{\mathrm{pw}}(x_i, x_j), \tag{2.15}$$

where $\mathcal{N}(i)$ are indices of the 4 neighbors of i. The condition $j > i$ ensures that we are not counting any neighboring pair twice. An MRF combining this prior with above likelihood is shown in Figure 2.3. Also notice how the discretized variational approach from Eq. (2.13)

---

[8] When we talk about translation-invariance, we disregard the fact that the finite size of the image will make this property hold only approximately.

is a special case of the generic pairwise MRF.

The question remains how we can choose an appropriate factor (or equivalently potential function) f_{pw} for the prior. In the majority of the cases in the literature, the potentials are hand-defined, resulting in an *ad hoc* model of image or scene structure. The resulting probabilistic models typically do not well represent the statistical properties of natural images and scenes. Because of the connection to variational approaches, this choice is directly linked to choosing an appropriate penalty function in the variational setup. Often, it is also assumed that the prior does not depend on the global gray level of the pixel pair; only their difference is considered, i. e.,

$$p(\mathbf{x}) = \frac{1}{Z_{\mathrm{pw}}} \prod_{i} \prod_{\substack{j \in \mathcal{N}(i), \\ j > i}} \tilde{f}_{\mathrm{pw}}(x_j - x_i). \tag{2.16}$$

This prior is formulated in terms of first derivatives, of which the neighbor difference is a simple approximation. But what is the right potential $\tilde{f}_{\mathrm{pw}}$?

Historically, these potentials have been modeled as Gaussians [Wong, 1968; Woods, 1972; Kashyap and Chellappa, 1981], i. e., $\tilde{f}_{\mathrm{pw}}(z) = e^{-z^2/(2\sigma^2)}$. But similar to the problems encountered in variational approaches (cf. Eq. (2.10)), this leads to image discontinuities being smoothed over. A number of other potential functions have been considered over the years, particularly potentials that are more tolerant toward large jumps in intensities as they occur at image discontinuities. If $\rho(z)$ is a robust penalty function, then the corresponding robust potential is given as $\tilde{f}_{\mathrm{pw}}(z) = e^{-c \cdot \rho(z/\sigma)}$, where c is a constant. There are many choices for the penalty function: One possibility is the truncated quadratic shown in Figure 1.4(a). With such a robust penalty function, this prior corresponds to the weak membrane model [Blake and Zisserman, 1987]. Other robust penalty functions are shown in Figures 2.2(b) and 2.2(c). In many cases, the form of the potential function is chosen by hand using simple considerations about the qualitative character of the data. Nevertheless in the task of modeling priors of images, it is possible to motivate the parametric forms of these robust potentials by studying the statistics of natural images (cf. Section 4.1 and [Tappen et al., 2003]); unfortunately this is rarely done. Moreover, until today the parameters of potentials are frequently chosen by hand in an ad-hoc fashion [Sun et al., 2003; Felzenszwalb and Huttenlocher, 2004]. Such hand-chosen models usually do not capture the statistics of our data of interest.

Learning and inference

Despite the fact that pairwise MRF models for low-level vision have largely remained hand-tuned, there exist a number of methods for learning these parameters from training data (see

[Li, 2001] for an overview). In the context of images, Besag [1986] uses the pseudo-likelihood criterion to learn the parameters of a parametric potential function of a pairwise MRF from training data. As mentioned above, maximum likelihood (ML) is probably the most widely used learning criterion for undirected graphical models in general, but it is computationally very demanding, because the partition function, which depends on the parameters of the MRF, cannot be computed in closed form. Nevertheless, ML estimation has been successfully applied to the problem of modeling images [Besag, 1974; Zhu and Mumford, 1997; Descombes et al., 1999]. ML estimation often relies on MCMC techniques to perform the necessary approximate inference steps.

Interestingly, even though the parameters of pairwise MRF models have largely remained hand-tuned, probabilistic inference has been much more widely adopted. Recently, efficient (approximate) inference algorithms such as belief propagation [Freeman et al., 2000; Sun et al., 2003; Tappen et al., 2003] and graph cuts [Boykov et al., 2001; Kolmogorov and Zabih, 2002] have been very widely used. Earlier on, iterated conditional modes [Besag, 1986], simulated annealing [Geman and Geman, 1984], and mean field methods [Geiger and Girosi, 1991] were used as well. The issue of choosing appropriate approximate inference algorithms for various vision problems was the subject of recent studies [Tappen and Freeman, 2003; Frey and Jojic, 2005; Szeliski et al., 2006].

In a number of applications, certain classes of potential functions are chosen to foster computational simplicity: The Potts model, for example, is often used in conjunction with graph cut algorithms [Boykov et al., 2001]. Felzenszwalb and Huttenlocher [2004] showed that for certain classes of penalty functions, such as truncated quadratics, messages in loopy BP can be computed in linear time instead of quadratic time using a distance-transform algorithm (linear in the number of labels, e. g., gray values, at each pixel). This is important, because BP can already be computationally intensive even for pairwise MRF models. While those models may make optimization easier, they unfortunately rarely capture the statistics of the problem and give the answer to the wrong problem (see also Figure 1.4). We recently showed that a larger class of robust potential functions can be approximated so that these distance-transform speedups are applicable as well [Lan et al., 2006], which alleviates these problems to some extent.

More complex pairwise MRFs

There are a number of pairwise MRF models that use more complex neighborhood systems beyond the 4-connected grid. Gimel'farb [1996] proposes a model with multiple and more distant pairwise neighbors, which is able to model more complex spatial properties that for example exist in textures (see also [Zalesny and van Gool, 2001]). Of particular note is that this method learns the neighborhood structure that best represents a set of training data. In the case of texture modeling, different textures result in quite different neighborhood systems. This work, however, has been limited to modeling specific classes of image texture and our experience with modeling more diverse classes of generic image structure suggests these methods do not scale well beyond narrow, class-specific, image priors. Even earlier work by Geman et al. [1990] formulated a pairwise MRF model for scene segmentation by adding a number of random long distance edges. In their model the long range edges did not necessarily improve on the kinds of local structures that can be represented, but helped disambiguating the locally ambiguous segment identifiers and aided convergence.

Problems of pairwise MRF models

Despite their long history and generality, pairwise MRF models have often produced disappointing results when applied to the recovery of complex scene structure, even when the parameters are learned. The denoising image in Figure 1.4(c) is characteristic of many MRF results; the robust potential function produces sharp boundaries but the result is piecewise smooth and does not capture the rich textural properties of natural scenes. This is in sharp contrast to results that have been obtained using methods that are tailored to a particular application, such as wavelet methods in image denoising [Portilla et al., 2003]. These techniques substantially outperform pairwise MRF models, but are not as widely applicable since they often do not immediately generalize to different applications.

For some years it was unclear whether limited application performance from pairwise MRFs was due to limitations of the model (i. e., from the pairwise structure or from unsuitable potentials), or due to limitations of the optimization approaches used for the non-convex MAP estimation problems. Meltzer et al. [2005] have recently obtained global solutions to low-level vision problems even with non-convex pairwise MRFs. Their results indicate that pairwise models are incapable of producing very high-quality solutions for stereo problems and suggest that richer models are needed for low-level modeling.

It is also worth noting that many other regularization approaches, for example variational [Schnörr et al., 1996] or nonlinear-diffusion related methods [Weickert, 2001], suffer from very similar limitations as many MRF approaches. Most frameworks penalize large image derivatives, which corresponds to choosing a very simple neighborhood system (the exact type depends on whether the terms involve the gradient magnitude or the individual derivatives). Moreover, in order to show the existence of a unique global optimum, many models restrict themselves to convex regularizers, but the missing connection to the statistics of natural images or scenes can be viewed as problematic. Also, non-convex regularizers often show superior performance in practice [Black et al., 1998]. There have

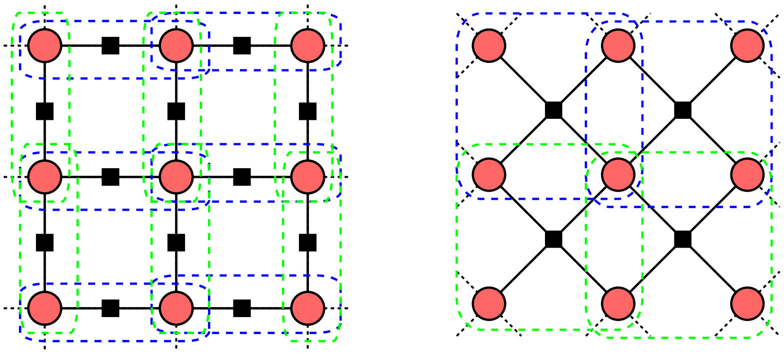

(a) Pairwise MRF (b) High-order MRF (2×2 cliques)

Figure 2.4: **Markov random field priors for low-level vision.** (a) The classical pairwise MRF (shown as factor graph) in which each node is connected to its 4 direct neighbors. (b) High-order MRF with larger cliques, here 2×2. The dashed, colored lines in each figure indicate the overlapping image patches that correspond to each shown factor.

been variational and diffusion-related approaches that try to overcome some of these limitations: Papenberg et al. [2006] penalize non-constancy of the gradient, but need to choose the trade-off parameters by hand. Gilboa et al. [2004] introduce a complex-valued regularization framework with a number of theoretical advantages, but the improvements in application performance remain minor.

2.2.3 High-order Markov random field models

There have been a number of attempts to go beyond these very simple pairwise MRFs that model the statistics of first derivatives in the image structure, and instead use high-order MRF models. High-order MRF models have already been suggested in the 1960s by Abend et al. [1965], but in most previous work they have been limited to the Gaussian case [Kashyap and Chellappa, 1981]. High-order MRFs with non-Gaussian potentials found more widespread use only in the 1990s [Geman et al., 1992; Zhu and Mumford, 1997; Zhu et al., 1998; Tjelmeland and Besag, 1998; Paget and Longstaff, 1998]. In a high-order MRF model of image structure, the maximal cliques (or the factors in the factor graph representation) encompass more than two pixels. Without loss of generality we assume the maximal cliques are based on square pixel patches of $m \times m$ pixels; other, non-square, neighborhoods can trivially be represented by having the potential function ignore pixels that are not in the neighborhood. Assuming there are K overlapping $m \times m$ patches in the image (plus corresponding factors), we can write such a high-order MRF prior abstractly as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{k=1}^{K} f(\mathbf{x}_{(k)}), \tag{2.17}$$

Figure 2.5: **Filters representing first and second order neighborhood systems** proposed by Geman and Reynolds [1992]. The left two filters correspond to first derivatives, the right three filters to second derivatives.

where $\mathbf{x}_{(k)}$ denotes the $m \times m$ patch associated with factor k. As in the pairwise case, such models typically assume spatial homogeneity; thus the factor (or potential) $f(\cdot)$ does not depend on the spatial location within the image (or other type of dense scene representation). Figure 2.4(b) shows the factor graph structure of a high-order MRF prior with factors defined on 2×2 patches. It is important to note here that the maximal cliques of such a high-order MRF and their corresponding patches overlap; there is a factor associated with every overlapping $m \times m$ patch in the data. This overlap is illustrated in Figure 2.4(b) as well, and as we can see in Figure 2.4(a), the overlap exists even in case of pairwise MRFs, where it may not have been as obvious.

As in the pairwise case, we have to choose an appropriate potential function $f(\cdot)$, which generally is more difficult in the high-order case because the potential is defined on larger cliques. A common approach involves modeling them based on higher-order derivatives of the image. For example Geman and Reynolds [1992] formulate a model where higher-order image derivatives (of order k) model more complex properties of the gray value surface of the image ($k = 1, 2, 3$ for piecewise constant, piecewise planar, or piecewise quadric images, respectively). Since image derivatives are linear operations, they can be implemented using linear filters. Figure 2.5 shows the 5 filters that are used to model piecewise planar images (here shown as square 3×3 filters) in the approach by Geman and Reynolds [1992]. The derivative responses are used to define the potential as follows:

$$f_{\mathrm{GR}}(\mathbf{x}_{(k)}) = \prod_{i=1}^{5} \exp\left(-c_i \cdot \rho(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)})\right). \tag{2.18}$$

The $\mathbf{J}_i$ are the five linear derivative filters shown in Figure 2.5, each stacked up into a column vector. The function ρ is a robust penalty function that penalizes large derivatives and thus encourages piecewise planar gray value surfaces; c_i is a weight that adjusts the influence of filter $\mathbf{J}_i$. We should also note here that if we only made use of the first two filters, this high-order model would be equivalent to the pairwise model from Eq. (2.16). Overall, it is thus a direct generalization.

Such a model is capable of representing richer structural properties beyond the piecewise spatial smoothness of pairwise models, but this and related models have remained largely hand-defined: The designer decides what might be a good model for a particular problem, chooses a neighborhood system and potential function without necessarily considering the statistics of the data, and typically tunes the parameters by hand. This is not only tedious, but also often only gives substandard results.

Other high-order MRF models in low-level vision

There have been a number of other high-order models used in low-level vision applications with the intent of modeling richer kinds of local structure. The Chien model for image segmentation [Descombes et al., 1999], for example, uses binary 3×3 cliques to model complex segmentation boundaries. The potentials are based on matching the local segmentation structure to a database of all possible binary 3×3 configurations (symmetric configurations are represented only once), each of which has a weight parameter associated with it that is learned from data. Extending the model beyond 3×3 cliques is difficult, because the number of possible clique configurations is very large. Tjelmeland and Besag [1998] propose a similar model on hexagonal grids, which is mostly used for binary segmentation. They prune the space of all possible binary configurations of each clique to keep the model tractable by manually identifying configurations that seem important. Furthermore, both models are specific to binary (or at least discrete) patterns and do not obviously generalize to continuous-valued data such as natural images.

To model high-order clique potentials a number of authors have turned to empirical probabilistic models captured by a database of image patches. Freeman et al. [2000] propose an MRF model that uses example image patches and a measure of consistency between them to model scene structure. This idea has recently been exploited as a prior model for image based rendering [Fitzgibbon et al., 2003] and super-resolution [Pickup et al., 2004]. There the clique potential of square cliques is defined using the minimum Euclidean distance to an example patch from the database (see Section 3.5.3 for more details). The roots of these models are in example-based texture synthesis [Efros and Leung, 1999], where non-probabilistic patch-based algorithms have shown impressive performance. These patch-based algorithms are not directly suited, however, as generic prior models for low-level vision. Example-based MRF models, on the other hand, are suitable as generic prior models and have lead to very good results [Fitzgibbon et al., 2003], but inference with these models is computationally intensive. This has inspired some of these authors to follow the approach developed in this dissertation [Woodford et al., 2006].

Example-based MRFs can be thought of modeling the clique potentials in a nonparametric way. In a similar vein Paget and Longstaff [1998] and Paget [2004] proposed a multiscale MRF model for texture synthesis in which the potentials are defined through multidimensional histograms.

The FRAME model

One model that is particularly important in the context of this dissertation is the FRAME model by Zhu et al. [1998]. It took a step toward more practical MRF models, as it is of high-order and allows MRF priors to be learned from training data, for example from a set of natural images [Zhu and Mumford, 1997]. The FRAME model defines clique potentials based on linear filter responses similar to the model by Geman and Reynolds, but with a greatly expanded repertoire of filters. Learning proceeds in several rounds by greedily choosing a filter from a bank of candidate filters using the minimum entropy criterion, and then adding it to the model. After selecting each filter, the parameters of the model are trained using the maximum entropy (or equivalently maximum likelihood) criterion. The bank of candidate filters is hand-chosen and mostly consists of Gabor filters of various orientations and scales. In addition to this, the model differs in two other important ways from the work by Geman and Reynolds: First, the filters are much larger, which implies that the model has much larger cliques. Secondly, the filter responses are modeled using a flexible discrete, nonparametric representation. This representation allows the model to match the marginal filter response statistics of the chosen filters to those of the data. The potential for clique $\mathbf{x}_{(k)}$ in the FRAME model can be written as

$$f_{\mathrm{FRAME}}(\mathbf{x}_{(k)}; \Lambda, \mathbf{J}) = \exp\left\{ -\sum_{i=1}^{N} \left(\sum_{l=1}^{L} \lambda_l^i \delta(z_l^i - \mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}) \right) \right\}, \tag{2.19}$$

where the $\mathbf{J}_i$ are the linear filters, the z_l^k are the positions of bins in a discrete histogram-like function, and λ_l^i are the respective function values. Λ is the set of all λ_l^i. The expression in parentheses can be thought of as a discrete representation of the robust penalty function in Eq. (2.18).

While the FRAME model has shown impressive results in the area of texture modeling [Zhu et al., 1997], the discrete nature of this representation complicates its use, and the reported image restoration results [Zhu and Mumford, 1997] appear to fall below the current state of the art. In contrast, the model that we will introduce in the following chapter uses parametric, differentiable functions to model filter responses. These potentials have direct computational advantages, because the differentiability facilitates various learning and inference methods. In particular, because we can differentiate the filter response model in the FoE framework, it is actually possible to learn the filters themselves from data.

2.2.4 Patch-based models

The statistics of small patches have received extensive treatment in the literature. Studies have mostly considered images, and not necessarily other types of scene representations in low-level vision. We will review patch-based models in the context of images, but note that many of these finding are likely to extend to other types of data, such as depth maps, etc. One of the most prominent approaches for modeling image patches is Principal Component Analysis (PCA) [Jolliffe, 2002], which yields visually intuitive components, some of which resemble derivative filters of various orders and orientations. The marginal statistics of such filters are highly non-Gaussian [Ruderman, 1994] (cf. Section 4.1) and are furthermore not independent, making PCA unsuitable for modeling image patches. Independent Component Analysis (ICA) [Bell and Sejnowski, 1995, 1997; Hyvärinen et al., 2003] instead assumes non-Gaussian statistics and finds the linear components that minimize the statistical dependence between them. As opposed to PCA, it yields localized components, which resemble Gabor filters of various orientations, scales, and locations. Since the components $\mathbf{J}_i \in \mathbb{R}^n$ found by ICA are by assumption independent, one can multiply their empirical marginal distributions $f_i(\mathbf{J}_i^{\mathrm{T}}\mathbf{x})$ to define a probabilistic model of image patches $\mathbf{x} \in \mathbb{R}^n$:

$$p(\mathbf{x}) \propto \prod_{i=1}^{n} f_i(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}). \tag{2.20}$$

It is very important to note that in contrast to our previous usage $\mathbf{x}$ here denotes a small, fixed-size image patch (for example, 15 × 15 pixels large). Also notice that projecting an image patch onto a linear component $(\mathbf{J}_i^{\mathrm{T}}\mathbf{x})$ is equivalent to filtering the patch with a linear filter described by $\mathbf{J}_i$. In case of image patches of n pixels it is generally impossible to find n fully independent linear components, which makes the ICA model only an approximation.

Even though typically motivated from an image-coding or neurophysiological point of view, there is a large amount of related work in the area of sparse coding, which attempts to address some of the issues in modeling complex image structure. Sparse coding methods [Olshausen and Field, 1996, 1997] represent an image patch in terms of a linear combination of learned filters, or “bases”, $\mathbf{J}_i \in \mathbb{R}^n$,

$$\arg\min_{\mathbf{J}} \min_{\mathbf{a}} E(\mathbf{a}, \mathbf{J}) = \sum_j \left\| \mathbf{x}^{(j)} - \sum_i a_{i,j}\mathbf{J}_i \right\|^2 + \lambda \sum_{i,j} S(a_{i,j}) \tag{2.21}$$

where $\mathbf{x}^{(j)} \in \mathbb{R}^n$ are vectorized image patches from the training data and $S(a_{i,j})$ is a sparseness prior that penalizes non-zero coefficients $a_{i,j}$. In the square or complete case, i. e., when the number of components equals the number of pixels, variations of this formulation lead to PCA and ICA as described above. The connection between analytical prior models, such as ICA, and synthesis-based models related to Eq. (2.21) is discussed in [Elad et al., 2006b].

Portilla et al. [2003] propose a sparse-coding related approach for image denoising. Their approach transforms the image using a sparsity-inducing overcomplete wavelet transform, and models correlations between a number of related wavelet coefficients. The approach models patches of wavelet coefficients based on a number of observations from natural image statistics, and does not give rise to a global model for all coefficients. Furthermore, the parameters of the model are inferred "online" (i. e., during denoising), which makes this model not directly suitable as a generic prior.

Products of Experts

Welling, Teh, and colleagues [Welling et al., 2003; Teh et al., 2003] went beyond the limitations of the square ICA model in Eq. (2.20) with a model based on the *Products-of-Experts* (PoE) framework [Hinton, 1999]. The idea behind the PoE framework is to model high-dimensional probability distributions by taking the product of several expert distributions, where each expert works on a low-dimensional subspace that is relatively easy to model. Usually, experts are defined on linear one-dimensional subspaces (corresponding to the basis vectors in sparse coding models). The projection onto these subspaces corresponds to filtering the image patch with the basis vector $\mathbf{J}_i$. Based on the observation that responses of linear filters applied to natural images typically exhibit highly kurtotic marginal distributions that resemble a Student t-distribution (cf. Fig. 1.3(a) and Section 4.1), Teh et al. [2003] propose the use of Student t-experts. The full Product of t-distribution (PoT) model can be written as

$$p(\mathbf{x}; \Theta) = \frac{1}{Z(\Theta)} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}; \alpha_i), \quad \Theta = \{\theta_1, \ldots, \theta_N\}. \tag{2.22}$$

The parameters $\theta_i = (\alpha_i, \mathbf{J}_i)$ consist of the expert parameters α_i and the associated linear filter $\mathbf{J}_i$. The Student t-experts ϕ have the form

$$\phi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}; \alpha_i) = \left(1 + \frac{1}{2}(\mathbf{J}_i^{\mathrm{T}}\mathbf{x})^2\right)^{-\alpha_i}. \tag{2.23}$$

The expert parameters α_i are assumed to be positive, which is needed to make the ϕ proper distributions (i. e., to ensure that their integral exists). $Z(\Theta)$ is a normalizing constant that ensures that the PoE density integrates to 1, but note that the experts themselves are not assumed to be normalized. It will later be convenient to rewrite the probability density in

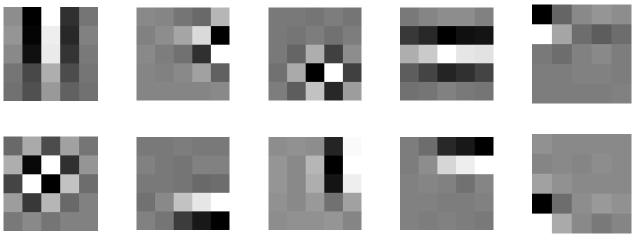


Figure 2.6: **Selection of the 5×5 filters** obtained by training the *Products-of-Experts* model on a generic image database.

Gibbs form as

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \exp(-E_{\mathrm{PoE}}(\mathbf{x}, \Theta)) \tag{2.24}$$

with

$$E_{\mathrm{PoE}}(\mathbf{x}, \Theta) = -\sum_{i=1}^{N} \log \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}; \alpha_i). \tag{2.25}$$

One important property of this model is that all parameters can be automatically learned from training data; that is both the expert parameters α_i and the linear filters $\mathbf{J}_i$. The advantage of the PoE model over the ICA model is that the number of experts N is not necessarily equal to the number of dimensions n (i. e., pixels). The PoE model permits fewer experts than dimensions (under-complete), equally many (square or complete), or more experts than dimensions (over-complete). The over-complete case is particularly interesting because it allows dependencies between filters to be modeled and consequently is more expressive than normal ICA.

We will not review the learning procedure for PoE models of image patches here, but remark that Teh et al. train them using contrastive divergence [Hinton, 2002]. A very similar learning procedure will be applied to the Fields-of-Experts model introduced in the following chapter. It is interesting to note that PoE models have a direct relation to projection pursuit methods for density estimation [Friedman et al., 1984]. But instead of greedily learning one projection at a time, PoE models train all projections (i. e., filters) at once. Figure 2.6 shows a selection of the 24 filters obtained by training this PoE model on 5×5 image patches[9]. The filters learned by this model are the same kinds of Gabor-like filters obtained using regular ICA techniques or standard sparse coding approaches. In the PoE framework, the shape of the t-distribution has the effect of a sparseness prior. Both for PoE and other sparse coding frameworks, it is possible to train models that are several times over-complete [Olshausen and Field, 1997; Teh et al., 2003]; the characteristics of the filters

---

[9] The training data contained about 60000 image patches randomly cropped from images in the Berkeley Segmentation Dataset [Martin et al., 2001]. The color images were converted to the YCbCr color space, from which we obtained gray scale versions by ignoring the chromatic channels Cr and Cb.

remain the same. The PoE framework has also successfully been applied to modeling of correlated wavelet coefficients [Gehler and Welling, 2006], while leading to state-of-the-art image denoising results.

Drawbacks and extensions

A key characteristic of these methods is that they focus on the modeling of small image patches rather than defining a prior model over an entire image. While this work provides possible insights into the neural coding of visual signals and suggests the use of particular filters, this work has not directly lead to practical algorithms for broad ranges of machine vision applications. Despite that, Welling et al. [2003] suggest an algorithm for denoising images of arbitrary size. The resulting algorithm, however, does not easily generalize to other image restoration problems such as image inpainting. Our focus here is not on any specific application such as denoising, but rather on finding a good general purpose framework for priors in low-level vision. As argued in the previous chapter, we view the aspect of modeling entire images (and other low-level scene representations) as very important in achieving this goal, and will hence pursue a different approach.

Some effort has gone into extending sparse coding models to full images [Sallee and Olshausen, 2003]. Inference with this model requires Gibbs sampling, which makes it somewhat less attractive for many machine vision applications.

Other work has integrated translation invariance constraints into the basis finding process [Hashimoto and Kurata, 2000; Wersing et al., 2003]. The focus there, however, remains on modeling the image in terms of a sparse linear combination of basis filters with an emphasis on the implications for human vision.

Other sparse-coding related methods

Beyond the methods already mentioned, there are other sparse-coding related approaches that instead attempt to model the properties of entire images in the context of image denoising, often using wavelet bases [Elad and Aharon, 2006; Lyu and Simoncelli, 2007]. While these approaches are motivated in a way that is quite different from Markov random field approaches in the pixel domain as emphasized here, they are similar in that they model the response to linear filters and some methods even allow the filters themselves to be learned. A key difference is that these models are not trained offline on a general database of natural images, but the parameters are instead inferred “online” in the context of the application at hand. While it is possible to extend these methods to other applications such as image inpainting [Mairal et al., 2006], doing so is relatively difficult due to the missing data in this case.

Other sparse-coding based formulations for entire images are more generically applicable to low-level problems. While still formulated in the context of image denoising [e. g., Elad et al., 2006a,b], these approaches are not trained "online" and are thus easier to generalize. Nonetheless, the performance is not competitive with above methods [e. g., Portilla et al., 2003].

2.2.5 Other models of low-level vision

Jojic et al. [2003] use a miniature version of an image or a set of images, called the epitome, to describe either image patches or a full image. Even though epitomes are generative models, they are usually built on a quite restricted class of images, even a single image. While it may be possible to use this method as a generic image prior by training it on a large and varied database of images, this possibility has not yet been explored.

The focus of this review has been on generative models, but even in discriminative approaches to low-level vision problems, it is necessary to model prior knowledge about the structure of images. We will not attempt to thoroughly review such approaches, but we at least want to mention two models that are interesting in the context of the proposed FoE model: He et al. [2004] propose a conditional random field model for image labeling that models complex structure in the label field using large cliques. The model is trained using a discriminative version of contrastive divergence, but as proposed is only applicable to binary data. Kumar and Hebert [2006] propose a general framework for learning discriminative random field models for labeling and classification tasks, where the labels are spatially dependent. While this work notes that large cliques may improve performance for these kinds of tasks, this possibility has not been explored.

CHAPTER 3

Fields of Experts

In this chapter we develop a framework for learning generic, expressive prior models for low-level vision that capture the statistics of natural scenes and can be used for a variety of machine vision tasks. The approach provides a practical method for learning high-order Markov Random Field (MRF) models with potential functions that extend over large pixel neighborhoods. These field potentials are modeled using a Products-of-Experts framework that exploits nonlinear functions of many linear filter responses. In contrast to previous MRF approaches this allows all parameters, including the linear filters themselves, to be learned from training data.

3.1 Introduction

While some of the models described in the preceding chapter provide an elegant and powerful way of learning prior distributions on small image patches, the results do not generalize immediately to give a prior model for whole images or for other dense scene representations[1]. Such a model is quite desirable, however, because as mentioned a substantial number of problems in low-level vision and image processing rely on a prior model of whole images, flow-fields, etc., for regularization. While a subset of these problems can be formulated using just a model of patches [e. g., Welling et al., 2003], a prior model of whole images (or some other dense scene representation) allows us to use a single framework for a large set of applications. Such global priors have been available for quite some time now, in particular through Markov random fields and related models [e. g., Geman and Geman, 1984], but the application performance of such models has (at least in some domains) fallen short of other methods that lack the generality and versatility of MRFs. Algorithms based on local,

---

[1] In this chapter we use the term "images" as the prototypical example of a dense scene representation of interest in low-level vision. Nonetheless, the descriptions are equally applicable to other scene representations such as flow-fields, dense scene depth, etc.

patch-based models are an example of this.

The question we may ask ourselves is how we can define and learn a global prior model for low-level vision, while using the lessons learned from modeling small patches? For several reasons simply making the patches bigger is not a viable solution: (1) the number of parameters to learn and store would be too large; (2) the model would only work for one specific image size and would not generalize to other sizes; and (3) the model would not be translation invariant, which is a desirable property for generic priors of low-level vision.

The key insight put forward here is that we can overcome these problems by combining ideas from modeling small patches with Markov random field models.

3.2 Basic Model

Before developing the FoE model, we should make clear what the data is that we are trying to model. Our goal in this chapter is to model the probability density of continuous-valued[2] images (or some other continuous-valued dense scene representation), which we denote as an M-dimensional random vector $\mathbf{x} \in \mathbb{R}^M$. Sometimes it will be convenient to think of the image as being arranged on its two-dimensional lattice of pixels, which in abuse of notation we denote as $\mathbf{x} \in \mathbb{R}^{r \times c}$, where r is the number of rows and c is the number of columns ($M = r \cdot c$).

As in a generic Markov random field model of low-level vision, we let the pixels of an image $\mathbf{x}$ be represented by nodes V in a graph $G = (V, E)$, where E are the edges connecting nodes. Our goal is to overcome the limitations of pairwise MRF models that typically rely on edges connecting each pixel to its 4 direct neighbors in the image grid [e. g., Geman and Geman, 1984; Black and Anandan, 1996; Sun et al., 2003]. Instead we define a high-order Markov random field using a neighborhood system that connects all nodes in an $m \times m$ square region. This is done for all *overlapping* $m \times m$ regions of the image. Every such neighborhood centered on a node (pixel) $k = 1, \dots, K$ defines a maximal clique $\mathbf{x}_{(k)}$ in the graph. By the properties of MRFs (cf. Section 2.1), we can write the probability density under this graphical model as a Gibbs distribution

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(-\sum_{k=1}^{K} U_k(\mathbf{x}_{(k)}) \right) = \frac{1}{Z} \prod_{k=1}^{K} f_k(\mathbf{x}_{(k)}), \tag{3.1}$$

where $\mathbf{x}$ is an image and $U_k(\mathbf{x}_{(k)}) : \mathbb{R}^{m \cdot m} \to \mathbb{R}$ is the potential function for clique $\mathbf{x}_{(k)}$. In

---

[2] Even though most of the exposition is based on continuous-valued images (or other similar, continuous-valued data), a substantial portion of the developed framework is independent of that assumption. We will point out important places where the developed framework does *not* immediately apply to discrete-valued data, such as segmentation masks, etc.

abuse of terminology, we will also refer to the factor $f_k(\mathbf{x}_{(k)}) : \mathbb{R}^{m \cdot m} \rightarrow \mathbb{R}^+$ as a potential. The term Z is a normalization term (cf. Eq. (2.5)), which ensures that the Gibbs distribution is properly normalized (i. e., integrates to 1). As is customary in most MRF models, we treat the boundaries of the image domain by only allowing $m \times m$ cliques that fully overlap with the image support. Alternatives are discussed in Section 3.6.

We make the additional assumption that the MRF is homogeneous; i. e., the potential function is the same for all cliques (or in other terms $f_k(\mathbf{x}_{(k)}) = f(\mathbf{x}_{(k)})$). This property gives rise to the translation-invariance of an MRF model. We should note that the translation invariance only holds approximately, because of the required boundary handling due to the finite size of the image.

Without loss of generality we usually assume that the maximal cliques in the MRF are square pixel patches of a fixed size. Other, non-square, neighborhoods could be used as well [cf., Geman and Reynolds, 1992], which will be discussed further in Section 4.5. Examples of such non-square cliques are shown in Figure 4.20, and include pairwise and diamond-shaped neighborhood systems. Such neighborhood systems can be treated as special cases of sufficiently large, square neighborhood systems by simply letting the potential function $f(\mathbf{x}_{(k)})$ ignore all pixels outside of the non-square shape.

As discussed in Section 2.2, potential functions $f(\cdot)$ of both pairwise and high-order MRFs have typically been hand-defined completely or at least to some extent. Our goal here is to learn the potential functions from training data and to keep the family of admissible potentials very general to allow the learning algorithm to find the most suitable potentials. The FRAME model by Zhu et al. [1998] was able to learn some of the parameters of the potential functions from data, but the candidate set of filters that was used to define the potentials was chosen by hand. In the model developed here, we want to learn the filters alongside its other parameters.

To enable that, we propose to represent the MRF potentials as a Product of Experts [Hinton, 1999] with the same basic form as in Eq. (2.22). This means that the potentials are modeled with a set of expert distributions that describe filter responses from a bank of linear filters. This global prior for low-level vision is a Markov random field of “experts”, or more concisely a *Field of Experts* (FoE). More formally, Eq. (2.22) is used to define the potential function (in factor form):

$$f(\mathbf{x}_{(k)}) = f_{\mathrm{PoE}}(\mathbf{x}_{(k)}; \Theta) = \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \boldsymbol{\alpha}_i). \tag{3.2}$$

Each $\mathbf{J}_i$ is a linear filter that defines the subspace that the corresponding expert $\phi(\cdot; \cdot)$ is modeling, and $\boldsymbol{\alpha}_i$ is its corresponding (set of) expert parameter(s). The matrix $\mathbf{J} =$

$(\mathbf{J}_1, \ldots, \mathbf{J}_N)$ is the matrix of all filter vectors. $\Theta = \{\mathbf{J}_i, \boldsymbol{\alpha}_i \,|\, i = 1, \ldots, N\}$ is the set of all model parameters. The number of experts and associated filters, N, is not prescribed in a particular way; we can choose it based on criteria such as quality of the model and computational expense. Since each factor can be unnormalized, we neglect the normalization component of Eq. (2.22) for simplicity.

Overall, the Fields-of-Experts model is thus defined as

$$p_{\mathrm{FoE}}(\mathbf{x}; \Theta) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K} f_{\mathrm{PoE}}(\mathbf{x}_{(k)}; \Theta) \tag{3.3}$$

$$= \frac{1}{Z(\Theta)} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \boldsymbol{\alpha}_i). \tag{3.4}$$

All components retain their definitions from above. It is very important to note here that this definition does not imply that we take a *trained* PoE model with fixed parameters Θ and use it directly to model the potential function. This would be incorrect, because the PoE model described in Section 2.2.4 was trained on independent patches. In case of the FoE, the pixel regions $\mathbf{x}_{(k)}$ that correspond to the maximal cliques are overlapping and thus not independent (this point is elaborated on in Section 3.5.1). Instead, we use the *untrained* PoE model to define the potentials, and learn the parameters Θ in the context of the full MRF model. What distinguishes this model from that of Teh et al. [2003] is that it explicitly models the overlap of image patches and the resulting statistical dependence; the filters $\mathbf{J}_i$, as well as the expert parameters $\boldsymbol{\alpha}_i$ must account for this dependence. It is also important to note that the FoE parameters $\Theta = \{\mathbf{J}_i, \boldsymbol{\alpha}_i \,|\, i = 1, \ldots, N\}$ are shared between all maximal cliques and their associated factors. This keeps the number of parameters moderate, because it only depends on the size of the maximal cliques and the number of experts, but not on the size of the image itself. Beyond that, the model applies to images of an arbitrary size and is translation invariant because of the homogeneity of the potential functions. This means that the FoE model can be thought of as a translation-invariant PoE model, which overcomes the problems we cited above.

Similar to the PoE (at least in its under- or overcomplete form) [Teh et al., 2003] and to most Markov random field models [Li, 2001], computing the partition function $Z(\Theta)$ of the FoE is generally intractable. One important fact to note is that the partition function depends on the parameters, Θ, of our model. Nevertheless, most inference algorithms, such as the ones discussed in Section 3.4, do not the require this normalization term to be known. During learning, on the other hand, we do need to take the normalization term into account, as we will see shortly.

We will frequently work with the log of the FoE model, and it is thus convenient to

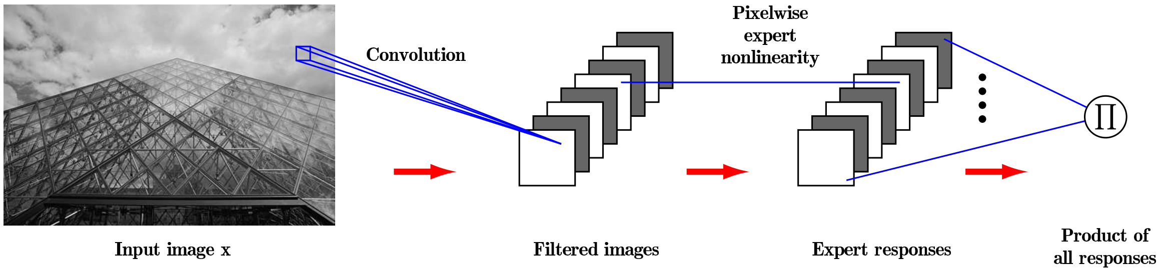


Figure 3.1: **Computational architecture of Fields of Experts.** The image $\mathbf{x}$ is convolved with a bank of linear filters. Each filter response image is fed into a corresponding expert nonlinearity. Finally the product is taken over all pixels of all expert response images, which results in a single scalar value - the unnormalized density $p_{\text{FoE}}(\mathbf{x};\,\Theta)$.

rewrite the model as

$$\begin{aligned} p_{\text{FoE}}(\mathbf{x};\,\Theta) &= \frac{1}{Z(\Theta)} \exp\left\{-E_{\text{FoE}}(\mathbf{x};\,\Theta)\right\} && (3.5)\\ &= \frac{1}{Z(\Theta)} \exp\left\{\sum_{k=1}^{K}\sum_{i=1}^{N} \psi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\,\boldsymbol{\alpha}_i)\right\} && (3.6)\\ &= \frac{1}{Z(\Theta)} \exp\left\{\sum_{k=1}^{K}\sum_{i=1}^{N} \psi_k(\mathbf{J}^{(i)} * \mathbf{x};\,\boldsymbol{\alpha}_i)\right\}, && (3.7) \end{aligned}$$

where log-experts are defined as $\psi(\cdot;\,\boldsymbol{\alpha}_i) = \log\phi(\cdot;\,\boldsymbol{\alpha}_i)$. In the last step, we assumed that we can also apply the log-experts elementwise; that is $\boldsymbol{\psi}(\mathbf{y})$ is the vector of all $\psi(y_k)$ and $\psi_k(\mathbf{y}) = \psi(y_k)$. This allows us to interpret the FoE in terms of convolution operations as shown in Eq. (3.7): The image $\mathbf{x}$ is convolved with each linear filter $\mathbf{J}^{(i)}$, the convolution result is fed into the pixelwise log-expert ψ_k, after which all results are summed and exponentiated to obtain the probability density under the model. The filter $\mathbf{J}^{(i)}$ is the convolution filter that corresponds to applying $\mathbf{J}_i$ to an image patch $\mathbf{x}_{(k)}$ [3]. Equivalently, we can also express Eq. (3.7) in terms of the expert functions themselves instead of the log-experts:

$$p_{\text{FoE}}(\mathbf{x};\,\Theta) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K}\prod_{i=1}^{N} \phi_k(\mathbf{J}^{(i)} * \mathbf{x};\,\boldsymbol{\alpha}_i). \qquad (3.8)$$

Here, we assume that the experts apply elementwise as well. Figure 3.1 illustrates this computational architecture. This formulation is not only interesting because it admits a slightly different interpretation of the model, but also from a practical point of view because it suggests that the (log-)density under the FoE model is very easy to compute.

---

[3] This requires mirroring the filter obtained by simply rearranging the vector into the columns of a filter matrix. This is necessary, because the convolution operation mirrors the filter mask before applying it to the image.

3.2.1 The experts

To make this general framework more specific, we have to choose appropriate expert functions $\phi(y; \boldsymbol{\alpha})$ for use in the FoE model; y here stands for the response to one of the linear filters. Similar to the PoE model, we have substantial freedom in doing so. The important criteria for choosing experts from a mathematical point of view are that the expert and its log are continuous and differentiable with respect to y and $\boldsymbol{\alpha}$; we will rely on these criteria during learning and inference (cf. Sections 3.3 and 3.4). From a modeling perspective, we want to choose experts that in the context of the full model give rise to statistical properties that resemble the data we want to model. As mentioned in the introduction, natural images and other dense scene representations have heavy-tailed marginal distributions, which motivates the used of heavy-tailed, highly kurtotic experts. We will discuss this choice in more detail in Section 4.2.

There are two experts that we are mainly considering here: (1) The very heavy-tailed Student t-distribution as it has been used in the PoE framework for modeling image patches (cf. [Teh et al., 2003] and Section 2.2.4); (2) A less heavy-tailed expert that is loosely based on the L1 norm, which has been successfully applied to a number of problems in image restoration [e. g., Donoho et al., 2006]. Other experts are certainly possible within this framework, but will remain the subject of future work[4].

In case of the Student t-expert, we take an unnormalized Student t-distribution [Gelman et al., 2004, §A.1]:

$$p_{\mathrm{t}}(y) \propto \left(1 + \frac{1}{\nu}\left(\frac{y-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}. \tag{3.9}$$

We assume the mean[5] μ to be 0, ignore the scaling of the random variable $1/(\nu \cdot \sigma^2)$, and replace the notion of the degrees of freedom ν with a single expert parameter α. We can safely ignore the scaling of the random variable, because it can be absorbed into the norm of each filter vector $\mathbf{J}_i$. This leads to the following Student t-expert:

$$\phi_{\mathrm{t}}(y;\, \alpha) = \left(1 + \frac{1}{2}y^2\right)^{-\alpha}. \tag{3.10}$$

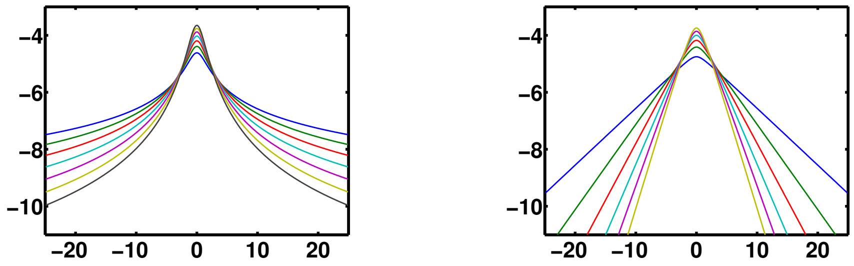
It follows that the log-expert and its derivative can be expressed as

$$\psi_{\mathrm{t}}(y;\, \alpha) = -\alpha \cdot \log\left(1 + \frac{1}{2}y^2\right) \tag{3.11}$$

---

[4] Also note that these experts are not necessarily suitable for use with discrete data $\mathbf{x}$, where the projections $y = \mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)}$ only take on values from a discrete set.

[5] Woodford et al. [2006] allow the mean to differ from 0 by extending each filter vector $\mathbf{J}_i$ by one element and extending each patch vector $\mathbf{x}_{(k)}$ with a constant of 1. We do not pursue this here, but note that this is an interesting direction to investigate in the future.



(a) Student t-expert.

(b) Charbonnier expert.

Figure 3.2: **Two expert functions used with the Fields-of-Experts framework** (log-plots).

and

$$\psi'_{\mathrm{t}}(y;\, \alpha) = -\frac{\alpha\, y}{1 + \frac{1}{2}y^2}. \tag{3.12}$$

The log-expert is illustrated in Figure 3.2(a). We will later also require the derivative of the log-expert w. r. t. the expert parameter α, which can easily be derived:

$$\frac{\partial}{\partial \alpha}\psi_{\mathrm{t}}(y;\, \alpha) = -\log\left(1 + \frac{1}{2}y^2\right). \tag{3.13}$$

In most of the applications shown in Chapters 4 and 5 this is the expert used.

Alternatively, we also use an expert based on the L1-norm (when viewed as energy), or equivalently the Laplacian distribution [Abramowitz and Stegun, 1964, §26.1]

$$p_{\mathrm{exp}}(y) \propto e^{-\frac{|y-\mu|}{\sigma}}. \tag{3.14}$$

We can once again assume the mean to be zero and absorb any scaling of the random variable into the norm of the filters $\mathbf{J}_i$. The problem is that this density and its log are not differentiable at zero, but we later require these properties. To remedy that, we instead use a “smoothed” version and define the following “Charbonnier” expert:

$$\phi_{\mathrm{C}}(y;\, \alpha, \beta) = e^{-\alpha\sqrt{\beta + y^2}}. \tag{3.15}$$

We fix the offset β to 1, but because y can be arbitrarily scaled through the filter norms, this incurs no loss of generality. The naming of this expert is derived from its (negative) log, which has been proposed by Charbonnier et al. [1994, 1997] for edge-preserving image restoration and was later used in conjunction with optical flow estimation [Bruhn et al., 2005]:

$$\psi_{\mathrm{C}}(y;\, \alpha) = -\alpha\sqrt{1 + y^2}. \tag{3.16}$$

This log-expert is illustrated in Figure 3.2(b). The derivative with respect to y is calculated as

$$\psi'_{\mathrm{C}}(y;\, \alpha) = -\frac{\alpha y}{\sqrt{1+y^2}}, \tag{3.17}$$

and the derivative w. r. t. the expert parameter α as:

$$\frac{\partial}{\partial \alpha}\psi_{\mathrm{C}}(y;\, \alpha) = -\sqrt{1+y^2}. \tag{3.18}$$

One aspect to note is that the Charbonnier expert is convex (more precisely its energy is convex). Because of that, FoE models with Charbonnier experts have a convex energy as well.

Common to both preceding log-experts is that their expert parameter is a scaling factor for a nonlinear response (cf. Eqs. (3.11) and (3.16)). Because of that, the expert parameter can be interpreted as a weight for a nonlinear response function.

3.2.2 Further properties

Before moving on to how we can learn the parameters of the FoE model from training data, we will explore a few mathematical properties that will later be useful.

For learning we will require the gradient[6] of the energy w. r. t. each expert parameter $\boldsymbol{\alpha}_i$. For the general case this is straight-forward to derive:

$$\nabla_{\boldsymbol{\alpha}_i} E_{\mathrm{FoE}}(\mathbf{x};\, \Theta) = -\sum_{k=1}^{K} \nabla_{\boldsymbol{\alpha}_i} \psi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i). \tag{3.19}$$

In this general form we only need to plug the respective gradient of the log-expert (e. g., Eq. (3.13) or Eq. (3.18)) into this equation. The resulting expression can be very efficiently implemented using convolutions.

The gradient of the energy with respect to a filter $\mathbf{J}_i$ will be required as well. It follows from applying the chain rule:

$$\nabla_{\mathbf{J}_i} E_{\mathrm{FoE}}(\mathbf{x};\, \Theta) = -\sum_{k=1}^{K} \psi'(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i) \cdot \mathbf{x}_{(k)}. \tag{3.20}$$

Once again, we only need to plug the appropriate expert derivative (e. g., Eq. (3.12) or Eq. (3.17)) into this expression. Compared to the parameter gradient, the filter gradient is slightly more tedious to implement. Apart from the usual convolutions, we also have to decompose the image into vectors of all its clique-sized patches (i. e., the $\mathbf{x}_{(k)}$).

---

[6] These derivations are carried out for the general case where each expert parameter $\boldsymbol{\alpha}_i$ could potentially be a vector. In the concrete cases we study, each expert only has a single parameter.

For most of our experiments in the subsequent chapters, we will parametrize the filter vectors in a different coordinate system denoted by the basis $\mathbf{A}$. As an example, such a filter basis can be used to define the filters in a space, where the clique-sized patches of the training data have unit covariance (i. e., are whitened). After writing the filters in the original space as $\mathbf{J}_i = \mathbf{A}^{\mathrm{T}}\tilde{\mathbf{J}}_i$, we can express the gradient of the energy with respect to the filters in the new coordinate system as

$$\nabla_{\tilde{\mathbf{J}}_i} E_{\text{FoE}}(\mathbf{x};\, \Theta) = -\sum_{k=1}^{K} \psi'(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i) \cdot \mathbf{A}\mathbf{x}_{(k)}. \tag{3.21}$$

Finally, we will require the gradient of the log-density w. r. t. to the image $\mathbf{x}$ itself[7]. We first rewrite the log-density in terms of a convolution as

$$\begin{aligned} \log p_{\text{FoE}}(\mathbf{x};\, \Theta) &= \sum_{k=1}^{K}\sum_{i=1}^{N} \psi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i) - \log Z(\Theta) \\ &= \sum_{k=1}^{K}\sum_{i=1}^{N} \psi_k(\mathbf{J}^{(i)} * \mathbf{x};\, \boldsymbol{\alpha}_i) - \log Z(\Theta). \end{aligned} \tag{3.22}$$

Then we calculate the partial derivative w. r. t. a pixel x_j

$$\frac{\partial}{\partial x_j} \log p_{\text{FoE}}(\mathbf{x};\, \Theta) = \sum_{k=1}^{K}\sum_{i=1}^{N} \psi'_k(\mathbf{J}^{(i)} * \mathbf{x};\, \boldsymbol{\alpha}_i) \cdot \frac{\partial}{\partial x_j}\left(\mathbf{J}^{(i)} * \mathbf{x}\right)_k, \tag{3.23}$$

where the rightmost expression is the filter response at pixel k, and the convolution filters $\mathbf{J}^{(i)}$ are defined as above. From this expression we can see that the filter coefficient that determines how strongly pixel j contributes to the filter response at pixel k also determines how strongly the log-expert derivative at pixel k contributes to the partial derivative of the log-density w. r. t. pixel j. Since we sum all the contributions from the various pixels k, this is equivalent to filtering the log-expert derivative with a filter that has been mirrored around its center; we call this filter $\mathbf{J}_{-}^{(i)}$. Overall, the gradient can thus be written as [cf., Zhu and Mumford, 1997]:

$$\nabla_{\mathbf{x}} \log p_{\text{FoE}}(\mathbf{x};\, \Theta) = \sum_{i=1}^{N} \mathbf{J}_{-}^{(i)} * \boldsymbol{\psi}'(\mathbf{J}^{(i)} * \mathbf{x};\, \boldsymbol{\alpha}_i). \tag{3.24}$$

Because this expression is based on convolutions, it is very simple and efficient to implement.

---

[7] This gradient is obviously only meaningful for continuous-valued data $\mathbf{x}$.

Figure 3.3: **Training data for FoE model of natural images.** Subset of the images used for the models described in Chapter 4. The training database contains images of animals, landscapes, people, architecture, etc.

3.3 Contrastive Divergence Learning

After establishing the FoE model we are ready to discuss how its parameters Θ, which include the expert parameters $\boldsymbol{\alpha}_i$ as well as the linear filters $\mathbf{J}_i$, can be learned from a set of training data $X = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(D)}\}$. In case of training an FoE model of natural images, the training data is a large set of natural images; Figure 3.3 shows example images from the image database used for training the FoE models in Chapter 4. For other types of dense scene representations, we need to collect appropriate training sets, which as we will see in Chapter 5 can be a challenge itself. Given the training data, we will attempt to estimate the parameters using the method of maximum likelihood (ML) [Geyer, 1991; Descombes et al., 1999]. This means that we are trying to maximize the following log-likelihood w. r. t. Θ:

$$\mathcal{L}(X;\,\Theta) := \log p_{\mathrm{FoE}}(X;\,\Theta) = \sum_{d=1}^{D} \log p_{\mathrm{FoE}}(\mathbf{x}^{(d)};\,\Theta). \tag{3.25}$$

As usual, we assume here that the data items are independent. One interesting property of ML estimation is that it is equivalent to minimizing the Kullback-Leibler (KL) divergence [MacKay, 2003, § 2.6] between the training data and the model distribution. This implies that we are trying to make the two distributions as close as we can (under the constraints of the model). While it is in principle possible to additionally put priors on the parameters themselves and perform MAP estimation, we will defer such a discussion until Chapter 4.

3.3.1 Maximum-likelihood estimation of FoE parameters

Unfortunately, the likelihood is generally highly non-convex, and in general, as for the PoE model, there is no closed form solution for the maximum likelihood parameters Θ_{ML}. Instead, we perform a gradient ascent on the log-likelihood of the data. For that we take the partial derivative of the log-likelihood with respect to a particular parameter θ_j (expert parameter or filter coefficient):

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\mathcal{L}(X;\Theta) &= \sum_{d=1}^{D}\frac{\partial}{\partial\theta_j}\log p_{\mathrm{FoE}}(\mathbf{x}^{(d)};\Theta) && (3.26)\\
&= \sum_{d=1}^{D}\frac{\partial}{\partial\theta_j}\Big(-E_{\mathrm{FoE}}(\mathbf{x}^{(d)};\Theta)-\log Z(\Theta)\Big) && (3.27)\\
&= -\left(\sum_{d=1}^{D}\frac{\partial E_{\mathrm{FoE}}(\mathbf{x}^{(d)};\Theta)}{\partial\theta_j}\right)-D\cdot\frac{\partial\log Z(\Theta)}{\partial\theta_j}. && (3.28)
\end{aligned}$$

Summing the contributions from all data items in the left term of Eq. (3.28) corresponds to taking an expectation w. r. t. the empirical distribution of X:

$$\sum_{d=1}^{D}\frac{\partial E_{\mathrm{FoE}}(\mathbf{x}^{(d)};\Theta)}{\partial\theta_j} = D\cdot\left\langle\frac{\partial E_{\mathrm{FoE}}(\mathbf{x};\Theta)}{\partial\theta_j}\right\rangle_X. \qquad (3.29)$$

The right term of Eq. (3.28) can be simplified as follows:

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\log Z(\Theta) &= \frac{\frac{\partial}{\partial\theta_j}Z(\Theta)}{Z(\Theta)} && (3.30)\\
&= \frac{1}{Z(\Theta)}\cdot\frac{\partial}{\partial\theta_j}\int\exp\{-E_{\mathrm{FoE}}(\mathbf{x};\Theta)\}\,d\mathbf{x} && (3.31)\\
&= \frac{1}{Z(\Theta)}\cdot\int\frac{\partial}{\partial\theta_j}\exp\{-E_{\mathrm{FoE}}(\mathbf{x};\Theta)\}\,d\mathbf{x} && (3.32)\\
&= \frac{1}{Z(\Theta)}\cdot\int-\frac{\partial E_{\mathrm{FoE}}(\mathbf{x};\Theta)}{\partial\theta_j}\exp\{-E_{\mathrm{FoE}}(\mathbf{x};\Theta)\}\,d\mathbf{x} && (3.33)\\
&= -\int\frac{\partial E_{\mathrm{FoE}}(\mathbf{x};\Theta)}{\partial\theta_j}p_{\mathrm{FoE}}(\mathbf{x};\Theta)\,d\mathbf{x} && (3.34)\\
&= -\left\langle\frac{\partial E_{\mathrm{FoE}}(\mathbf{x};\Theta)}{\partial\theta_j}\right\rangle_{p_{\mathrm{FoE}}(\mathbf{x};\Theta)}. && (3.35)
\end{aligned}$$

Here, we can swap the partial derivative and the integration, because the integrand is continuous and differentiable. Eq. (3.35) implies that computing the partial derivative of the partition function requires taking an expectation w. r. t. the model distribution $p_{\mathrm{FoE}}(\mathbf{x};\Theta)$.

With these intermediate results we express the partial derivative of the log-likelihood as

$$\frac{\partial}{\partial \theta_j} \mathcal{L}(X;\, \Theta) = D \cdot \left[- \left\langle \frac{\partial E_{\mathrm{FoE}}(\mathbf{x};\, \Theta)}{\partial \theta_j} \right\rangle_X + \left\langle \frac{\partial E_{\mathrm{FoE}}(\mathbf{x};\, \Theta)}{\partial \theta_j} \right\rangle_{p_{\mathrm{FoE}}(\mathbf{x};\, \Theta)} \right]. \tag{3.36}$$

We can thus perform a gradient ascent on the log-likelihood by updating each parameter θ_j by

$$\delta \theta_j = \eta \left[- \left\langle \frac{\partial E_{\mathrm{FoE}}}{\partial \theta_j} \right\rangle_X + \left\langle \frac{\partial E_{\mathrm{FoE}}}{\partial \theta_j} \right\rangle_{p_{\mathrm{FoE}}} \right], \tag{3.37}$$

where η is a user-defined learning rate. This basic methodology for ML estimation is common to a number of different techniques including the Product of Experts and other energy-based models [Teh et al., 2003]. But while large parts of the above derivations parallel those in [Teh et al., 2003], there are subtle, important differences discussed in Section 3.5.1.

The expectation over the training data is fortunately easy to compute, but on the other hand there is no general closed form solution for the expectation over the model distribution, just as is the case for the PoE. This means that we have to approximate this expectation, for example using Monte Carlo techniques: We can approximately compute the expectation by repeatedly drawing samples from $p_{\mathrm{FoE}}(\mathbf{x};\, \Theta)$ [see e. g., MacKay, 2003, § 29] based on the following approximation:

$$\langle g(x) \rangle_{p(x)} \approx \frac{1}{S} \sum_{s=1}^{S} g(x^{[s]}), \tag{3.38}$$

where the samples $x^{[s]}$ are drawn from $p(x)$. Since we cannot draw samples from the FoE model distribution directly, we have to rely on Markov Chain Monte Carlo (MCMC) techniques for sampling (see e. g., [Andrieu et al., 2003] or [Gelman et al., 2004, § 11]). This strategy has been applied to ML learning in general [Geyer, 1991], and to ML learning in MRFs in particular [Zhu et al., 1998; Descombes et al., 1999].

MCMC techniques are very powerful methods, but they often have extreme computational demands. Complex models, such as the FRAME model [Zhu et al., 1998], can take weeks to train on current computers. This stems from two facts: (1) The dimensionality of the space in which we have to sample is quite large. To train the FoE we have to sample *entire images* with many pixels. This can be mitigated by using relatively small images, but is still a significant burden as MCMC techniques are often inefficient in high dimensions. The reason for why we have to sample entire images, and not just small image patches that have the same size as the cliques, is that we need to capture the overlap between the clique patches in the image; (2) MCMC techniques often require many iterations until the Markov chain (approximately) converges to the target distribution.

There have been a number of attempts at increasing the efficiency of such a learning procedure. One possibility is to make MCMC sampling more efficient using intelligent, data-driven proposal distributions [e. g., Tu and Zhu, 2002] or domain-specific proposal distributions, such as the Swendsen-Wang method [Barbu and Zhu, 2005]. Another possibility is to avoid having to do maximum likelihood estimation altogether by using a very different learning objective, such as maximum pseudo-likelihood[8] [Besag, 1974], maximum "satellite" likelihood [Zhu and Liu, 2002], or score matching [Hyvaärinen, 2005]. We follow a similar path here, but exploit a very different learning criterion called *contrastive divergence* (CD), which was proposed by Hinton [2002] and has already been successfully applied to a number of related models such as the PoE [Teh et al., 2003], as well as conditional random fields [He et al., 2004; Kumar and Hebert, 2006]. To our knowledge, the Field-of-Experts model is the first application of contrastive divergence to learning generative MRF models.

3.3.2 Contrastive divergence applied to Fields of Experts

Instead of making a major change to the learning rule developed above, contrastive divergence takes a simple, but nevertheless powerful shortcut: Instead of running the Markov chain for computing the expectation over the model distribution until convergence, we only run it for a small, fixed number of iterations, l, while starting the Markov chain at the training data[9]. We could use as few as $l = 1$ MCMC iteration, which is what we use for training FoE models if not noted otherwise. We use these samples in place of the "true" converged samples that would ordinarily be used to approximate the learning rule from Eq. (3.37). If we denote the data distribution as p^0 (the state of the Markov chain after 0 iterations) and the distribution after l MCMC iterations as p^l_Θ, the contrastive divergence learning rule thus updates the parameters θ_j using

$$\delta\theta_j = \eta \left[-\left\langle \frac{\partial E_{\text{FoE}}}{\partial \theta_j} \right\rangle_{p^0} + \left\langle \frac{\partial E_{\text{FoE}}}{\partial \theta_j} \right\rangle_{p^l_\Theta} \right]. \tag{3.39}$$

This learning rule has both an intuitive as well as a mathematical interpretation. The intuition is that initializing the Markov chain at the data makes sure that we have samples in all places where we want the model distribution to be accurate, and that running the MCMC sampler for just a few iterations starting from the data distribution already draws the samples closer to the (current) model distribution. This change is sufficient to estimate the parameter updates.

Mathematically, Hinton [2002] showed that updating the parameters according to Eq. (3.39)

---

[8] Pseudo-likelihood methods are not easily applicable here, because the form of the FoE model makes computing the necessary conditionals difficult.

[9] Strictly speaking, we have one Markov chain for each training datum.

is (almost) equivalent to following the gradient of

$$D(p^0 || p_\Theta^\infty) - D(p_\Theta^l || p_\Theta^\infty), \tag{3.40}$$

where $D(\cdot||\cdot)$ denotes the KL divergence, and p_Θ^∞ denotes samples from the converged Markov chain. The interpretation is that instead of minimizing the KL divergence directly as in the maximum likelihood case, we are trying to minimize the change in KL divergence from performing l MCMC iterations (hence the name of the learning algorithm). This means that we are minimizing the tendency for samples to move away from the training data and by that make the model distribution closer to the data distribution. Note that we are ignoring a typically very small term in this interpretation [see Hinton, 2002].

There have been a number of theoretical analyses of contrastive divergence. For certain distributions and under certain conditions, it can be shown that CD is in fact equivalent to ML estimation [Williams and Agakov, 2002; Yuille, 2005]; for others it can be shown that CD finds different parameters than ML estimation [MacKay, 2001]. Most recently, Carreira-Perpiñán and Hinton [2005] showed that CD is biased in many cases, but that the bias is very small in practice for important classes of models.

3.3.3 Sampling strategies

There are various kinds of sampling schemes that can be used to do MCMC sampling in the context of contrastive divergence, including advanced ones such as the methods mentioned above. A classical method often used in the context of modeling images is the Gibbs sampler [Geman and Geman, 1984], which repeatedly samples subsets of the random variables by keeping the other variables fixed. In many image-related applications, the Gibbs sampler is used to sample each pixel conditioned on all other pixels [e.g. Geman and Geman, 1984; Zhu et al., 1998], but this can lead to slow mixing of the Markov chain. Another possibility is to augment the random vector $\mathbf{x}$ to be sampled using a set of auxiliary variables $\mathbf{u}$; these auxiliary variables are then sampled conditioned on $\mathbf{x}$ and vice versa. Such an alternating Gibbs sampler can be quite efficient and has, for example, been used with Products of Experts [Welling et al., 2003] or with Restricted Boltzmann Machines [Hinton, 2002]. In the case of a PoE with Student t-experts (but not limited to this case), this augmentation arises from an interpretation as a Gaussian scale mixture model (see also Section 3.6). The FoE architecture allows for a very similar procedure, but that is not explored here.

In our implementation, we use a hybrid Monte Carlo (HMC) sampler (see e. g., [Neal, 1993; Andrieu et al., 2003] for detailed descriptions), which is another efficient sampling technique, typically much more efficient than the frequently used Metropolis sampler. The advantage of the HMC sampler stems from the fact that it uses the gradient of the log-density to explore the space more effectively. In analogy to a physical particle system, the random vector $\mathbf{x}$ is augmented with a "momentum" term $\mathbf{m}$ that describes the (differential) motion of the particle at $\mathbf{x}$ through space. The energy of the target distribution $p_{\text{FoE}}(\mathbf{x}; \Theta)$ is thought of as an external field that over time changes the initially random momentum to direct the particle toward low-energy, i. e., high-probability areas. This is done through the gradient of the log-probability, which for the FoE model is given in Eq. (3.24). With this method a particle can traverse the space quite quickly, which results in a fast mixing Markov chain. This "particle motion" is discretized in practice using so-called leapfrog steps with some step size; we use 30 such steps in our implementation. Because the discretization of this continuous process is only approximate, HMC corrects this approximation using a Metropolis acceptance step; we adjust the leapfrog step size adaptively so that the acceptance rate is near 90%. This sampling strategy was also used by Teh et al. [2003] in the context of modeling image patches with Products of Experts.

There is an interesting interpretation of this sampling process in the context of images. The random momentum $\mathbf{m}$, which is drawn from a Gaussian distribution, corresponds to an image of white noise. The particle starts at some image $\mathbf{x}$ from the training database (because CD initializes the sampler at the data), and over time is influenced by the momentum, which makes the image noisier and thus less like a "natural" image. This tendency to make the image noisier is counteracted by the gradient of the log-density, which biases the momentum toward real images (according to the current model). As we will see in more detail in Section 4.3, this gradient can be thought of as smoothing the image, i. e., removing the initial noise. This thus builds an intuitive connection to applications of the FoE model as will be studied in the subsequent chapters.

3.3.4 Implementation details

In order to correctly capture the spatial dependencies of the overlapping neighboring cliques (or equivalently the overlapping image patches), the size of the images (or other spatial data that we want to model) in the training data set should be substantially larger than the clique size. On the other hand, large images would make the required MCMC sampling inefficient. We train this model on 20000 randomly cropped image regions that have at least 3 times the width and height of the maximal cliques (i. e., in case of 5×5 cliques we train on 15×15 images). Instead of using the entire dataset at each iteration, we split the data set into "mini-batches" of 200 images and use only the data from one batch at each iteration. This procedure is called stochastic gradient ascent (see [Bottou, 2004] for an overview). While we still consider the entire training data set, it takes 100 iterations of the learning algorithm to sweep through the entire data set. These mini-batches lead to a considerable speedup in training time. In most of our experiments, we use 5000 iterations with a learning rate of $\eta = 0.01$. We found the results to not be very sensitive to the exact value of the learning rate.

Despite that, training FoE models using contrastive divergence is a computationally very intensive task. Training a model with 8 filters of size 3 × 3 and image data of size 15 × 15 takes about 8 CPU hours on a single, modern PC (Intel Pentium D, 3.2 GHz). Training a 5 × 5 model with 24 filters takes about 24 CPU hours to complete. It is important to note here that training occurs offline; that is the FoE prior is trained once ahead of time, and can then be used in a variety of applications (see Chapter 4).

Since the parameter gradient is "noisy" because of the sampling procedure as well as the stochastic gradient ascent, we use a momentum term [cf., Teh et al., 2003] to stabilize the ascent. If n denotes the current iteration, then we update parameter θ_j according to

$$\delta\theta_j^{(n)} := \nu \cdot \delta\theta_j^{(n-1)} + (1-\nu) \cdot \delta\theta_j, \tag{3.41}$$

where $\delta\theta_j$ is determined as in Eq. (3.39) and $\delta\theta_j^{(0)} = 0$. This averages recent parameter updates by weighting them according to how recently they occurred. In our experiments ν is set to 0.9. The fact that the gradient is "noisy" also makes it difficult to establish automatic convergence criteria. We thus manually monitor whether the model parameters have stabilized sufficiently.

The experts introduced in Section 3.2.1 require their expert parameters α to be positive for them to be valid probability densities. While it is possible to define a proper overall probability density under the FoE model (i. e., one whose integral exists) without requiring all experts to be valid densities themselves[10], in our experiments we found no considerable advantage from allowing such improper experts. Hence, we ensure positivity of the α parameters during learning by updating them in the log-domain:

$$\alpha_i^{(n)} := \exp\left\{\log(\alpha_i^{(n-1)}) + \eta\alpha_i^{(n-1)}\delta\alpha_i^{(n-1)}\right\}. \tag{3.42}$$

Here we use the fact that $\frac{d}{d\log\alpha} g(e^{\log\alpha}) = \alpha \cdot \frac{d}{d\alpha} g(\alpha)$. Updating the log also seems to help stabilizing the learning procedure.

As mentioned above, we sometimes define the filters $\mathbf{J}_i$ in a different coordinate system such that $\mathbf{J}_i = \mathbf{A}^{\mathrm{T}}\tilde{\mathbf{J}}_i$, and learn the transformed filters $\tilde{\mathbf{J}}_i$ using the gradient from Eq. (3.21). After learning, we obtain the filters in the original space by transforming the learned filters.

---

[10] Some of the constrains in the minimax entropy framework by Zhu et al. [1997, 1998] take the form of improper densities; i. e., their associated energies are "reversed" and have a global *maximum*. The "overcompleteness" of this and also the FoE model (see Section 3.6) makes it possible for the overall density to still be proper.

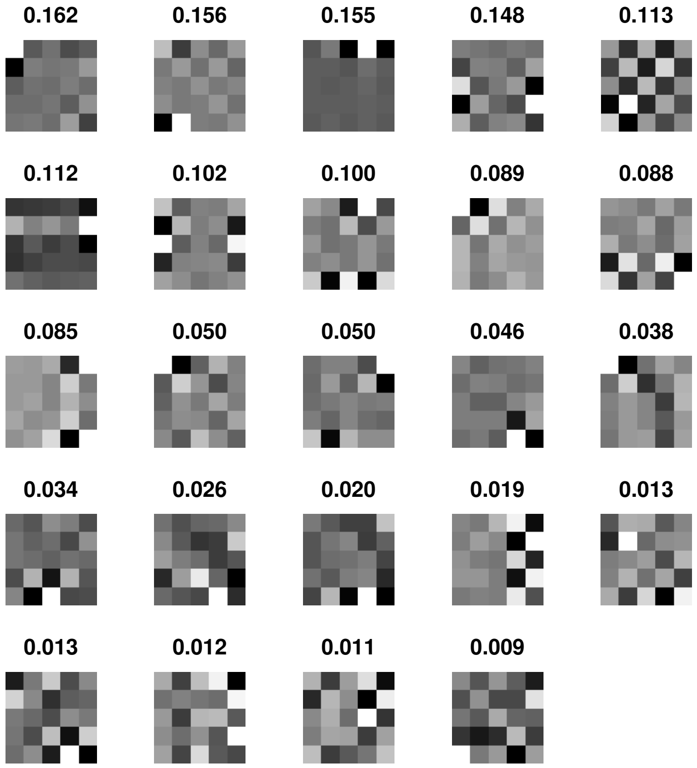


Figure 3.4: **Learned** 5×5 **filters** obtained by training the *Fields-of-Experts* model with Student t-experts on a generic image database (see Section 4.2 for details). The number above each filter denotes the corresponding expert parameter α_i.

It is important to note that this does not change the maximum likelihood objective, if the linear transformation $\mathbf{A}$ has full rank. In other terms, the global optima of the data likelihood are still the same (when the filters are viewed in the original space). Nonetheless, since the stochastic gradient-based learning algorithm only finds local optima, this may still change the local optima that are found. As we will discuss in more detail in Section 4.5, we find that encouraging the filters to exhibit high-frequency components substantially improves application performance. The transformation $\mathbf{A}$ we typically use is motivated as follows: First, let $\mathbf{\Sigma}$ denote the covariance of independent training data patches of the same size as the filters. Then we can decompose the covariance using an eigendecomposition as $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{\Lambda}$ is diagonal and $\mathbf{U}$ is orthonormal. The transformation $\mathbf{W} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$ whitens the image patches $\mathbf{y}$, so that $\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y}$ has unit covariance. This also means that the low-frequency components have the same variance as the high-frequency components. In order to encourage high-frequency components, we need to raise their variance and lower the variance of the low-frequency components. We can achieve this by defining an "inverted" whitening transformation $\mathbf{A} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$, which simply inverts the eigenvalues.

In some of the applications of the FoE model, it may be advantageous to make it invariant to additive constants, i. e., $p_{\mathrm{FoE}}(\mathbf{x}) = p_{\mathrm{FoE}}(\mathbf{x} + c \cdot \mathbf{1})$, where $\mathbf{1}$ is a vector of all ones[11]. In case of modeling images, additive constants are tantamount to global shifts in brightness. We can achieve this invariance, for example, by restricting the basis $\mathbf{A}$ in which the filters $\mathbf{J}_i$ are defined so that the filter coefficients of $\mathbf{A}^{\mathrm{T}}\mathbf{J}_i$ have mean 0. If we do not use such a basis, we can simply remove the mean from the training data beforehand and from the filters $\mathbf{J}_i$ after each iteration of the learning procedure. Figure 3.4 shows the filters and the associated expert parameters for a 5×5 FoE model with 24 Student t-experts that was trained on a database of natural images (see Section 4.2 for more details). The model was trained by encouraging high-frequency filters using the transformation from above, and removing the mean from consideration.

3.4 Inference with the FoE Model

After training Fields of Experts using contrastive divergence, we can use them in a number of different low-level vision applications. These applications rely on probabilistic inference, which includes the problem of finding maximum a-posteriori (MAP) solutions and the problem of computing marginal distributions of the posterior. But before actually describing these applications in the subsequent chapters, we will discuss the general issue of probabilistic inference with the FoE model. While all experiments in this dissertation are based on MAP inference, computing marginals may be important for future applications and is thus reviewed here as well. For this generic inference case we will assume that our goal is to recover $\mathbf{x}$ based on a measurement $\mathbf{y}$ using an FoE prior $p_{\mathrm{FoE}}(\mathbf{x})$ and an application specific likelihood $p(\mathbf{y}|\mathbf{x})$. The posterior is then written as

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) \cdot p_{\mathrm{FoE}}(\mathbf{x}). \tag{3.43}$$

In particular, we will describe how the techniques for inference in general graphical models described in Section 2.1.1 apply to FoE-based models. As discussed there, exact inference in graphical models with loops is very difficult and even approximate inference is computationally very expensive. As we will see below, this is even exacerbated by the fact that FoE models are based on large cliques. One important thing to keep in mind is that we do not need to know the normalization term $Z(\Theta)$ to perform inference. This is because the parameters Θ are fixed during inference and most inference methods (including all methods mentioned below) only need to know the posterior up to some arbitrary constant.

Inference with FoE models can be done in two different ways: (1) We can retain the

---

[11] We should note that this constraint makes the density under the FoE model improper, which is normally not a major problem in practice.

continuous-valued interpretation of the FoE framework, and do inference in the continuous domain; (2) We can abandon the continuous-valued interpretation and discretize the state of each node in the graphical model, which is quite easily possible for many vision-related applications. The typical $[0, 255]$ range of gray values is often discretized in steps of 1 gray level. After doing this, we can approach the problems of inference in the discrete domain, where a large part of recent work on inference in graphical models has focused.

Sampling-based methods Sampling from the posterior distribution is a very attractive way of inference, because it not only allows us to compute approximate MAP solutions and marginals, but also expectation values of global functions, which are useful for certain applications. Most of the sampling techniques discussed in Section 3.3.3 in the context of learning are also applicable to the problem of probabilistic inference in Fields of Experts. Such sampling methods mostly rely on the original continuous-valued FoE model and do not require a discretization of the state. The limitation here is that the images in actual applications are much larger than the ones used for training, which can render MCMC samplers too inefficient to be practical. Nonetheless, there have been recent successes even in complex vision applications [Tu and Zhu, 2002; Barbu and Yuille, 2004]. Once we have such a sampling algorithm available, they can also be used inside a simulated annealing algorithm for MAP estimation [Geman and Geman, 1984]. We do not follow any sampling approaches here, but note that it would be very interesting to investigate the use of advanced sampling techniques such as data-driven MCMC [Tu and Zhu, 2002] or the Swendsen-Wang method [Barbu and Zhu, 2005] in conjunction with FoE models.

Graph cuts MAP estimation using graph cuts [Boykov et al., 2001] has recently been gained a lot of attention in computer vision [Tappen and Freeman, 2003; Boykov and Kolmogorov, 2004; Szeliski et al., 2006]. In the general case, the complexity of these methods is exponential in the size of the maximal cliques; that is if S is the number of discrete states in an FoE with $m \times m$ cliques, then the complexity of graph cut-based inference is $\mathcal{O}(\text{poly}(S^{m \cdot m}))$. Because of that, graph cuts have to our knowledge not been applied to models with cliques of size 5×5 or even 3×3. The application of graph cuts for inference with FoE models (or more precisely their discrete-valued approximation) would be an interesting avenue for future work. In particular, it seems worth investigating if the particular architecture of the FoE model allows to reduce the computational effort.

Message passing Apart from graph cuts, message passing techniques, especially loopy belief propagation [Yedidia et al., 2003], have been very influential for inference in graphical models, including models of low-level vision. Loopy BP and variants are interesting in the context of the FoE model, because they not only allow approximate computations of MAP solutions, but also of marginal distributions. While most applications of loopy BP are based on simple pairwise MRF models, BP can also be derived for arbitrary factor graphs [Kschischang et al., 2001; Yedidia et al., 2005], which means that BP can be directly applied in the context of the FoE.

Unfortunately, for discrete models message computations are computationally very expensive especially in high-order graphs. In general, the complexity is exponential in the clique size [Lan et al., 2006]: If S is again the number of discrete states of each node and $m \times m$ cliques are used in the FoE, then computing each message takes $\mathcal{O}(S^{m \cdot m})$ operations in the general case. In other work, we have explored how these message computations can be approximated to make them tractable in practice [Lan et al., 2006], but so far the resulting algorithm is still limited to FoE models with 2×2 cliques. Because of that, we do not follow this approach for the experimentation in this dissertation. It is highly desirable, however, to extend this work and enable BP to be used with high-order graphical models such as the FoE. In particular, it seems worthwhile to investigate whether the structure of the FoE potentials can be exploited to speed up message computation, possibly related to the distance transformation techniques proposed for pairwise graphs [Felzenszwalb and Huttenlocher, 2004]. Potetz [2006] is currently exploring such speed-ups.

Belief propagation with continuous models has so far been limited to simple Gaussian MRFs (and thus is not applicable here), or relied on nonparametric message representations [Sudderth et al., 2002; Isard, 2003]. Even though they have proven quite useful, they do not seem easily applicable in the context of inference in large, high-order models, because of the computational expense of message computation.

A separate family of message passing algorithms for continuous graphical models has been developed based on messages represented by distributions in the exponential family. These techniques are known as expectation propagation [Minka, 2001, 2005]. While they are, in principle, applicable to FoE-based models, it is not clear which exponential family distribution would be suitable for representing the messages, as they would likely need to be multimodal. Further research in this direction seems to be a fruitful area for future work.

Continuous optimization Finally, it is possible to retain the continuous interpretation of the FoE model and to maximize the (log) posterior using techniques from nonlinear-optimization. The simplest such method is gradient ascent, where we simply follow the gradient of the log-posterior. This is done based on the following ascent rule:

$$\mathbf{x} \leftarrow \mathbf{x} + \tau \left[\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \nabla_{\mathbf{x}} \log p_{\text{FoE}}(\mathbf{x}) \right], \tag{3.44}$$

where τ is a suitable stepsize. The contribution of the (log) prior $\nabla_{\mathbf{x}} \log p_{\text{FoE}}(\mathbf{x})$ can easily be computed using Eq (3.24), which makes this inference algorithm very easy to apply and very easy to implement. Conjugate gradient methods [Fletcher, 1987] typically lead to faster convergence than simple gradient ascent and are applicable to the problem of MAP estimation in FoE models as well. In particular, we can use the Polak-Ribière variant, for which widely used, robust implementations are available [Rasmussen, 2006]. Of course, since the energy of the model is often highly non-convex, both techniques will typically only find local optima of the objective. An exception to this is an FoE based on Charbonnier experts, which has a convex energy and is thus easy to optimize with gradient techniques. But as we will see in the subsequent chapters, gradient-based optimization methods can lead to very good results even for non-convex energies.

3.5 Comparison to Related Models

There are a number of techniques that the Field-of-Experts model as introduced is related to. In this section, we will discuss the similarities and differences between the FoE and related approaches in more detail.

3.5.1 Product of Experts

Since the FoE is directly derived from the Product of Experts [Hinton, 1999], the relationship to the PoE is quite close and worth understanding thoroughly. On a first glance the models look formally very similar: The FoE model is given as (cf. Eq. (3.3))

$$p_{\text{FoE}}(\mathbf{x};\, \Theta) = \frac{1}{Z_{\text{FoE}}(\Theta)} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i), \tag{3.45}$$

whereas the PoE model is given as (cf. Eq. (2.22))

$$p_{\text{PoE}}(\mathbf{x};\, \Theta) = \frac{1}{Z_{\text{PoE}}(\Theta)} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x};\, \boldsymbol{\alpha}_i). \tag{3.46}$$

The key formal difference is that in the FoE we take a product over all patches of an image, whereas the PoE models only a single patch. But what happens if we simply assume independence and multiply together PoE models for all patches $\{\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(K)}\}$ in an image?

We obtain

$$\begin{aligned} p_{\text{PoE}}(\{\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(K)}\};\, \Theta) &= \prod_{k=1}^{K} p_{\text{PoE}}(\mathbf{x}_{(k)}) && (3.47) \\ &= \prod_{k=1}^{K} \frac{1}{Z_{\text{PoE}}(\Theta)} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i) && (3.48) \\ &= \frac{1}{Z_{\text{PoE}}(\Theta)^K} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i). && (3.49) \end{aligned}$$

Apart from the normalization term, this equation appears identical to Eq. (3.45) from above. So how does the FoE model differ from this product of PoE models?

The answer lies in exactly this normalization term. The PoE is normalized by considering the distribution on a single patch, i. e.,

$$Z_{\text{PoE}}(\Theta) = \int \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x};\, \boldsymbol{\alpha}_i)\, d\mathbf{x}. \tag{3.50}$$

The FoE on the other hand is normalized by considering the distribution of many *overlapping* patches

$$Z_{\text{FoE}}(\Theta) = \int \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)};\, \boldsymbol{\alpha}_i)\, d\mathbf{x}. \tag{3.51}$$

This means that in general

$$Z_{\text{FoE}}(\Theta) \neq Z_{\text{PoE}}(\Theta)^K, \tag{3.52}$$

and because of that the maximum likelihood parameters of the FoE and the PoE model will generally not be the same. When viewed from the perspective of the learning algorithm, this difference means that to train a PoE model we sample *independent patches*, but to train an FoE model we sample independent images consisting of *dependent patches*. This explains why it is important that we sample images that are larger than the cliques when training an FoE.

Because the patches overlap, it is incorrect to assume independence and multiply together independent, trained PoE models as in Eq. (3.47). In the FoE on the other hand, we multiply the untrained experts on overlapping patches and then train their parameters so that the overlap and thus the dependence is properly accounted for. This means that the expert parameters $\boldsymbol{\alpha}_i$ and the filters $\mathbf{J}_i$ of the FoE model have to account for this overlap.

3.5.2 FRAME model and maximum entropy

The FRAME model [Zhu et al., 1997; Zhu and Mumford, 1997; Zhu et al., 1998] is also directly related to the FoE model put forward here, but motivated in a different way. The prior probability density of an image $\mathbf{x}$ under the FRAME model can be written in our notation as (cf. Eq. (2.19))

$$p(\mathbf{x};\Lambda,\mathbf{J}) = \frac{1}{Z(\Lambda)} \exp\left\{ -\sum_{k=1}^{K} \sum_{i=1}^{N} \left(\sum_{l=1}^{L} \lambda_l^i \delta(z_l^i - \mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}) \right) \right\}, \tag{3.53}$$

where the z_l^k are the positions of bins in a discrete function and λ_l^i are the respective function values. Λ is the set of all λ_l^i. In analogy to the FoE formulation, the term in parentheses can be thought of as an expert. The key difference is that the experts in the FoE framework are parametric, continuous functions, whereas in the FRAME model they are nonparametric, discrete functions. Because the experts are discrete functions in case of the FRAME model, we cannot directly compute derivatives w. r. t. their argument, and thus we cannot directly compute the gradient of the energy with respect to the filters $\mathbf{J}_i$ (cf. Eq. (3.20)). This means that the filters cannot be learned as it is done in conjunction with the FoE model. Because of that, Zhu et al. [1997] propose a hand-designed candidate set of linear filters and choose filters from this set using a greedy search method.

Liu et al. [2001] propose a variant of the FRAME model for the much lower-dimensional domain of face modeling. In their model, the discrete feature functions are replaced by a continuous nonparametric function based on Parzen windows, which allows for differentiation with respect to the projection vectors (comparable to the filters here). This can be used for a greedy search for appropriate projection directions. In other work, we have applied this methodology to the joint modeling of neural firing in monkey motor cortex and of concurrently observed physical behavior [Wood et al., 2006]. Training such a model is quite tedious, however, even in such relatively low-dimensional spaces. Furthermore, the search process suffers quite severely from local optima. Because of that, applying the same methodology to models of images does not seem feasible.

The motivation for the FRAME model comes from the maximum entropy principle [e. g., Jaynes, 2003]; that is it models the prior distribution of images or specific textures by maximizing the entropy of the model subject to certain constrains. This makes it as uncommitted as possible aside from the imposed constraints. It can be shown that maximum entropy models (also called maxent, or ME) take the form of a Gibbs distribution with linearly weighted features. The FRAME model imposes that the responses to the hand-defined filters $\mathbf{J}_i$ have the same marginal distribution under the model as they do on the training data [Zhu et al., 1998]. This directly leads to the formulation in Eq. (3.53). Certain types of FoE models can be understood in the context of maximum entropy as well. In particular, if the log-expert can be written as

$$\psi(y;\, \alpha_i) = \alpha_i \cdot \xi(y), \tag{3.54}$$

which is possible for both the Student t- and the Charbonnier expert, then we can rewrite the FoE in form of a maxent model:

$$p_{\mathrm{FoE}}(\mathbf{x};\, \Theta) = \frac{1}{Z(\Theta)} \exp\left\{ \sum_{i=1}^{N} \alpha_i \cdot \left(\sum_{k=1}^{K} \xi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}) \right) \right\}. \tag{3.55}$$

This means that the FoE (for such experts) is a maxent model that constrains the expectation of the sum of nonlinearly transformed filter-responses $\sum_{k=1}^{K} \xi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)})$ to be the same on the training data and under the model distribution.

An interesting direction of future work would be to find expressive expert functions for use in the FoE, which are differentiable, but have a flexibility similar to the discrete experts of the FRAME model. Such a model could be more powerful than a model based on simple parametric experts, while still keeping the ability to learn the filters.

3.5.3 Other high-order MRF models

As discussed in Section 2.2, there have been various other attempts at modeling low-level vision data using high-order models. Most notable in this context the work by Geman and Reynolds [1992] and Geman et al. [1992], which uses high-order cliques that promote locally linear or quadric structures, as opposed to locally constant structures that are encouraged by typical pairwise MRF models. The mathematical setup is related to the FoE model, but relies on simple derivative filters of different orders. The nonlinearity in their formulation takes the form of

$$\phi_i(y) = e^{-c_i \rho(y)} = \exp\left\{ \frac{c_i}{1 + |y|} \right\}. \tag{3.56}$$

The key difference is that the model is fully hand-defined; both the filters as well as the weights of the nonlinearities are chosen by hand. The FoE model as presented here extends this work by allowing for these parameters to be learned from data.

A number of techniques have used example data to model potential functions of high-order MRF models themselves, so called example-based priors [Freeman et al., 2000; Fitzgibbon et al., 2003; Pickup et al., 2004]. For example in [Fitzgibbon et al., 2003], clique potentials for 5×5 cliques are defined as

$$f(\mathbf{x}_{(k)}; \mathcal{T}) = \exp\left\{-\lambda \min_{\mathbf{t}\in\mathcal{T}} ||\mathbf{t} - \mathbf{x}_{(k)}||_2^2\right\}, \tag{3.57}$$

where $\mathcal{T}$ is a large database of example image patches. This potential is based on the Euclidean distance of the clique patch $\mathbf{x}_{(k)}$ to the closest patch in the example database, $\mathbf{t}$. In contrast, the potentials in the FoE model are fully parametric and while examples are used to train the model, they are not part of the representation. While example-based priors lead to very good results, inference in such a model is quite difficult and slow. Continuous optimization techniques are, for example, not applicable to example-based priors. Recent followup work [Woodford et al., 2006] has replaced the example-based prior with a Field of Experts in the context of the same application, which resulted in qualitatively similar results but considerably faster inference.

3.5.4 Convolutional neural networks

Another quite interesting interpretation of the FoE model is that of a neural network [Bishop, 1995], or more specifically of a convolutional neural network (see [LeCun et al., 1998] for an overview). Figure 3.1 essentially illustrates this architecture (when viewed as energy-based model): The image builds the input layer ($M \approx K$ nodes) and the convolutions are the weights that connect the input layer to the hidden layer of the nonlinearly transformed filter responses. If N filters are used, the hidden layer has $N \cdot K$ nodes. Finally, the output layer has a single node without a nonlinearity; a weight vector of all ones sums all the contributions from the hidden layer. Because the weights that connect the input layer to the hidden layer are shared between all spatial locations in the image in a convolutional fashion, such a network is also called a convolutional neural network.

Nonetheless, there is a key difference between traditional neural networks and the FoE model. Neural networks are chiefly used in the discriminative setting for regression and classification tasks. For these applications, they are traditionally trained using back-propagation, though a fully Bayesian treatment is possible [Neal, 1996]. The FoE model on the other hand is a generative model corresponding to the general density estimation problem, and thus needs to be trained in generative fashion. If we want to use the FoE architecture in this generative way, then there are relatively few alternatives to ML or MAP estimation (cf. Section 3.3). But it is possible to use similar architectures as in the FoE in a discriminative fashion. Ning et al. [2005], for example, propose a convolutional neural network for segmentation, which uses large “cliques” to model complex spatial structures. The model is trained in a discriminative fashion using the energy-based learning framework laid out by LeCun and Huang [2005].

3.6 Discussion

Before concluding this chapter and moving on to applications of the FoE model to various problems in low-level vision, there are a few more aspects of the model that are worth discussing. One particularly interesting aspect is the nature of the filters that are obtained by training the FoE model. If we look at Figure 3.4, we can see that the filters learned by the model have high spatial frequency and are seemingly irregular, but also look rather different than the filters obtained by training a Product of Experts on image patches (cf. Fig. 2.6). They are also very different from the filters obtained by ICA [Bell and Sejnowski, 1997] or by other sparse-coding methods [Olshausen and Field, 1996], which resemble oriented derivative filters of various orientations, scales, and locations. Yet, the filters learned by the FoE model look nothing like ordinary derivative filters. Even if we do not encourage high-frequency filters using the transformation **A** defined above, the learned filters still have high frequency and appear rather irregular. Nevertheless, if we discourage the learning algorithm from finding high-frequency filters, the model performs substantially less well in applications. Moreover, if we do not learn the filters, but instead use a fixed set of filters in the FoE, such as the random ones, or the filters obtained by ICA or PoE, the model performs less well in applications. This suggests that the nature of the filters is important, even if they may look rather irregular. More details on these findings are given in Section 4.5. For now, we will try to understand what could make the filters differ from those obtained by patch-based models.

Understanding the learned filters. As we have already explored in the previous section, the FoE differs from the PoE in that it models overlapping, dependent image patches. Nevertheless, the FoE takes the form of a model that promotes independence of filter responses. The easiest way to understand this is to compare the FoE model to independent component analysis (ICA) [Bell and Sejnowski, 1995; Hyvärinen and Oja, 2000]. ICA maximizes the independence between components (i. e., linear projections) of the data using a variety of different criteria. One such criterion is maximum likelihood, in which case the ICA model is written as [cf., Hyvärinen and Oja, 2000, Eq. (31)]

$$p_{\mathrm{ICA}}(\mathbf{x}; \mathbf{W}) \propto \prod_{i=1}^{N} f_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}), \tag{3.58}$$

where $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_N)^{\mathrm{T}}$ is the so-called “unmixing matrix” and $f_i(\cdot)$ are the assumed densities of the independent components. Note that this is a special case of the PoE framework, in particular a complete PoE model. This means that in this ICA framework independence is encouraged by multiplying together the density of the components and finding the projections $\mathbf{w}_i$ that maximize the likelihood under Eq. (3.58). In the FoE model we multiply together the unnormalized expert densities across all filters *and* all clique locations. This means that maximum likelihood estimation of the FoE parameters encourages independence of the filter responses, but not only between different filters, but also between responses of the same filter at different image locations. This independence between filter responses at different image locations poses additional constraints on the shape of the filters, and may explain why the filters differ from a patch-based model such as ICA or PoE. Fields of Experts can thus be thought of as a translation-invariant PoE model.

It may seem counterintuitive that even though we know that the patches corresponding to the cliques are dependent, we still use a formulation that encourages independence of filter responses. Nevertheless, as Williams [2005] shows this can still be a sensible thing to do: He studies an autoregressive model for sequences and shows that the random variables of the sequence can be transformed in a way that makes them look independent [cf., Williams, 2005, Eq. (2.6)]. In particular, he suggests a simple transformation based on differences of neighboring variables ("difference observations") and notes that the transformed features can be treated as a Product-of-Experts model. He goes on to suggest that a similar transformation ought to be possible for spatial models. This is a strong motivation for the FoE model in that it shows that we can model spatial dependence using "independent" features. In order for that to work, the features, or in other terms the filters, have to be chosen appropriately. The flexibility of the FoE model allows us to learn filters that correspond to such "difference observations", here of higher order. Moreover, this interpretation gives us another way of trying to understand the nature of the filters.

Completeness. As we saw in Section 2.2.4, the Product-of-Experts model can have a varying number of experts and associated linear projections (i. e., filters), and depending on the number of experts the model could be complete, under-complete, or over-complete. How do these attributes apply to Fields of Experts, since there the number of filters is also variable? If we disregard effects from the image boundaries, then even an FoE model with one filter is already complete, because every clique has an expert and there are (approximately) as many cliques as there are pixels. All models with two or more filters are over-complete as they have more experts (per entire image) than pixels.

Boundary handling. In the FoE model as we defined it above every overlapping $m \times m$ patch of the image (or other dense scene representation) forms a clique or factor in the MRF. But what about pixels at or near the boundary? In the given formulation, we only associate cliques with patches that fully overlap the image, which means that pixels at or near the boundary appear in fewer cliques than pixels in the interior. The effect is that they are less constrained and can vary more than pixels in the interior. Boundary handling is an old problem in image processing and low-level vision [Horn, 1986] and unfortunately, there are no completely satisfactory solutions. Circular boundary conditions, where the image is thought of as the surface of a torus, work well for very structured data such as textures, but for generic images, they would introduce artificial image edges. Implicitly assuming that all pixels outside of the image support are zero introduces similar effects. Reflective boundary conditions or replicating the boundary outside the support seem more appropriate in the context of modeling generic scene representations, but can get quite tricky to implement. Because of that, we do not follow these approaches, but note that they may constitute good ways of improving the performance at boundaries. Instead, we continue to assume that only fully overlapping patches have an associated clique or factor.

This has one drawback: Since the boundary pixels are constrained by fewer cliques, there appears to be a certain bias in the learning procedure to give large filter weights to coefficients near the boundary of the filter, because those will be responsible for modeling the pixels at the boundary of the image. There are two possible ways of reducing this bias. One is to increase the size of the patches used for training, in order to minimize the relative importance of the boundary pixels, but this makes training computationally much more expensive. The other approach is to train the FoE model in a conditional fashion, e. g.,:

$$p_{\mathrm{FoE}}(\mathbf{x}_{\mathrm{int}}|\mathbf{x}_{\mathrm{ext}};\ \Theta) = \frac{1}{Z(\Theta, \mathbf{x}_{\mathrm{ext}})} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)};\ \boldsymbol{\alpha}_i). \tag{3.59}$$

Here, $\mathbf{x}_{\mathrm{int}}$ are the “interior” pixels (i. e., pixels away from the boundary), and $\mathbf{x}_{\mathrm{ext}}$ are the “exterior” pixels on and near the boundary. This can easily be done in practice by choosing the exterior region just large enough so that each pixel in the interior is covered by equally many cliques, and only sampling the interior region during contrastive divergence learning. While this changes the characteristics of the filters somewhat as reported in Section 4.5, this procedure does not have a substantial effect on application performance.

Connection to Gaussian scale mixtures. Many heavy-tailed distributions can be expressed as so-called Gaussian scale mixtures (GSM) [Andrews and Mallows, 1974; Barndorff-Nielsen et al., 1982]. In particular, the zero-mean density $p(x)$ can in this case be expressed as an infinite mixture of Gaussians with mean zero, but different variance:

$$p(x) = \int_0^\infty p(x, z)\, dz = \int_0^\infty \mathcal{N}(x;\ 0, z\sigma^2) \cdot p(z)\, dz, \tag{3.60}$$

where $\mathcal{N}(x;\ 0, z\sigma^2)$ is a zero-mean Gaussian distribution whose variance is adjusted according to the scale z, and $p(z)$ is the prior density over the scales. Distributions that can be written in this way include the Student t-distribution, the Laplacian distribution, which is similar to the "Charbonnier" expert used here, as well as several other popular ones that are relevant in the context of modeling natural image statistics [Wainwright and Simoncelli, 2000]. As already mentioned, this property has been exploited by Welling et al. [2003] in the context of the PoE to define an efficient Gibbs sampling procedure.

For example in the case of Student t-experts (and zero mean filters), we can write the FoE model as a GSM by formulating each expert as a GSM and combining the results:

$$p_{\mathrm{FoE}}(\mathbf{x};\ \Theta) = \int_{\mathbf{z}} \mathcal{N}\left(\mathbf{x};\ \mathbf{0}, \mathbf{F}(\mathbf{J}) \cdot \operatorname{diag}\{\mathbf{z}\} \cdot \mathbf{F}(\mathbf{J})^{\mathrm{T}}\right) \cdot \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{k,i};\ \alpha_i)\, d\mathbf{z}, \tag{3.61}$$

where $z_{k,i}$ are independent scale variables for each filter and every location, and $\mathbf{F}(\mathbf{J})$ is a large, sparse matrix formed by convolution operators for all filters $\mathbf{J}_i$. We omit additional details, as we make no further use of this connection, but note that it can, for example, be used for efficient Gibbs sampling.

Discriminative modeling. In Section 3.3 we discussed how FoE models can be trained in a generative fashion using contrastive divergence. Nonetheless, the FoE model architecture may also prove useful for discrimination tasks. To that end the FoE could be recast as a conditional random field (CRF) model [Lafferty et al., 2001], where the experts model the spatial dependencies of the output variable $\mathbf{x}$, but use the input $\mathbf{y}$ to adapt the local potentials. Instead of maximizing the likelihood, training such a model could be performed using conditional likelihood maximization. The multiscale conditional random field by He et al. [2004] is essentially an instantiation of such a conditional variant of the FoE. Furthermore, such a conditional variant would strongly resemble the discriminative random field model by Kumar and Hebert [2006], which is set up in a similar fashion, but models spatial interactions only through pairwise terms.

Measuring the quality of FoE models. To conclude our discussion, we should note that assessing the quality of FoEs is quite difficult in general. This is because we cannot compute the normalization term $Z(\Theta)$ for the FoE model, which means that we cannot use the likelihood of either training data or a separate test set to determine whether the training data is well modeled and whether the model potentially overfits the training data. In absence of the normalization term, we have to rely on indirect measurements of the quality of the FoE model. Instead we use the trained FoE in context of a vision application and judge the quality of the model from the performance on this application. In the next chapter we analyze the performance of various FoE models on the tasks of image denoising and image inpainting. A number of different aspects of the FoE are evaluated with regards to their impact on application performance. Also, these experiments allow us to compare the FoE to other models of low-level vision. In Chapter 5 we measure the performance of a number of FoEs on the task of optical flow estimation.

3.7 Summary

In this chapter we introduced Fields of Experts, a flexible framework for modeling prior distributions of various types of low-level vision representations. Even though the framework was introduced mainly for modeling continuous-valued data such as natural images, optical flow, depth maps, etc., a substantial part of the discussion equally applies to discrete-valued data such as segmentations. The FoE model overcomes the limitations of patch-based models by making them translation invariant and applicable to scene representations of an arbitrary size. On the other hand, the FoE extends previous Markov random field models using spatially extended cliques, which are modeled using nonlinear expert functions that operate on linear filter responses. We suggested two different expert models to be used in the framework, and showed how the FoE can be trained using contrastive divergence. In contrast to previous high-order MRFs, the linear filters can be learned from training data as well. We discussed the issue of probabilistic inference, and even though a number of recently very popular inference algorithms such as belief propagation and graph cuts are currently infeasible for FoEs, approximate inference can be done using a very simple gradient ascent procedure. We finally discussed the differences and similarities of the FoE model to related approaches, and pointed out a number of details that further motivate the approach.

CHAPTER 4

Modeling Natural Images

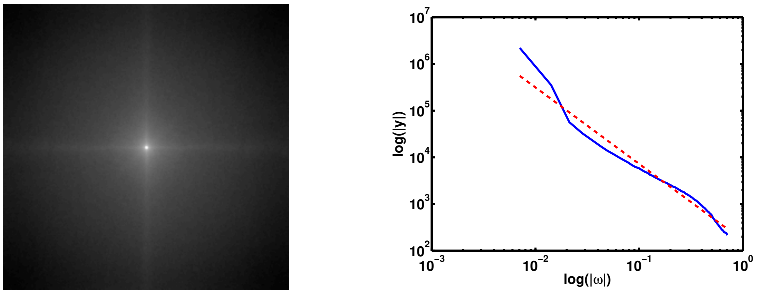
In this chapter we study the application of Fields of Experts to the problem of modeling natural images. First, we will review the key statistical properties of natural images. Following that, we will present a particular FoE model for modeling natural images, and show how it is trained on a comprehensive image database. We will also discuss a number of important observations about the FoE in order to better understand the model. In the next part, we will study two different applications in which an FoE model of natural images can be used, namely image denoising and image inpainting. For image denoising we will demonstrate extensive experimental results and a comparison to the current state-of-the-art. The image inpainting application will mainly serve to illustrate the versatility of an FoE prior of natural images; we will only show a few key results. The last part of this chapter will present an extensive study of various properties of the Fields-of-Experts model, including, for example, how the number of experts influences the performance.

4.1 Natural Image Statistics

Before we can review and study the statistics of natural images, we should first clarify what me mean by *natural images*, as multiple conflicting definitions exist. Some studies, for example, distinguish between natural and man-made scenes; there “natural” only refers to anything that is part of nature as opposed to made by humans. We take a more general view, and define natural images as the kinds of images that one could typically encounter in real life. This broader category includes landscapes, animals, and people, but also architecture and other man-made objects, as well as people in such environments[1]. There are a variety of image databases that have been made available over the past decade. Many of these

---

[1] A potentially more precise way of characterization is to call this class of images *photographic images*, a term that has for example been used by Simoncelli [2005]. Since this term has not been that widely adopted, we will still mostly refer to this class as “natural images”.



(a) (b)

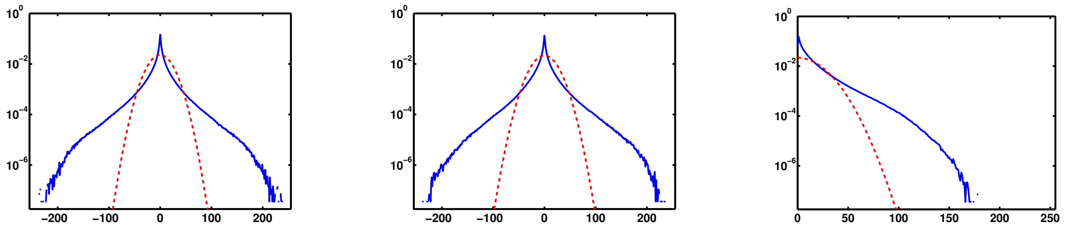
Figure 4.1: **Spectra of natural images.** (a) 2D spectrum. The slight horizontal and vertical lines indicate a strong presence of horizontal and vertical structures. (b) Power spectrum (solid blue) and simple fit with a power-law distribution (dashed red). The data approximately follows a $1/|\omega|^{1.64}$ power-law.

databases have a substantial focus on images from nature [van Hateren and van der Schaaf, 1997] and are thus not appropriate for our definition of natural images. We use the image collection that accompanies the Berkeley segmentation dataset [Martin et al., 2001], which itself is a subset of the Corel image database. The dataset is comprised of 200 training images from a wide variety of scenes both from nature and everyday life. Figure 3.3 shows a few example images from this database. All studies in the remainder of this chapter are carried out on these images after converting them to gray scale. We obtained gray scale versions by converting the data to the YCbCr color space and ignoring the chromaticity Cr and Cb, and retaining the luminance component Y. Note that unlike other works [e. g., Huang, 2000], we do not use logarithmic intensities here, but instead work with normal, gamma-compressed intensity images (see below for further discussion).

4.1.1 Key properties

The study of natural images dates back quite a long time and is by now comprised of a very large set of publications. We will only review the most important aspects that are relevant for modeling prior distributions of natural images with Fields of Experts. Readers who are interested in more detailed studies are referred to the works of Field [1987]; Ruderman [1994, 1997]; Huang [2000]; Lee et al. [2001]; Grenander and Srivastava [2001]; Lee et al. [2003]; Srivastava et al. [2003]; and Simoncelli [2005].

Power-law spectra. As early as the 1950s, television engineers studied the statistics of television signals [Kretzmer, 1952]. They discovered that in television images the energy for a particular (spatial) frequency f follows a power-law, i. e., it is proportional to $1/|\omega|^{2-\epsilon}$. These properties have been (re-)discovered in the context of natural images by Field [1987]



(a) Horizontal derivative ($\kappa = 19.27$)

(b) Vertical derivative ($\kappa = 17.78$)

(c) Gradient magnitude

Figure 4.2: **Marginal log-histograms of natural images (solid blue).** The Gaussian distribution with the same mean and variance is shown as well (dashed red). The kurtosis of the derivative histograms is given in the respective figure.

and others. Figure 4.1 shows the spectral properties of the image database used here[2]. As we can see from the slight horizontal and vertical lines in Figure 4.1(a), there is a considerable occurrence of horizontal and vertical image structures, which are quite frequent in man-made environments. Also, we can see how the power spectrum is fit relatively well by a power-law distribution ($\epsilon = 0.36$). Nonetheless, frequency properties of images are rarely used in their modeling, because they are global properties, whereas typical models of natural images including the FoE are localized in some form.

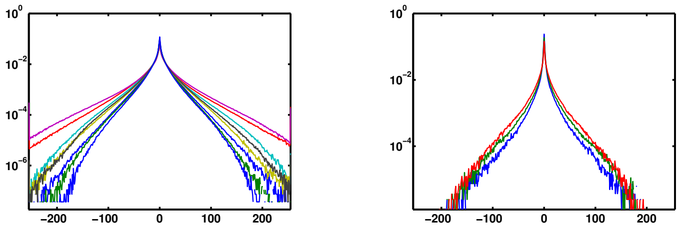
Marginal statistics. More important for real applications was the discovery of heavy-tailed marginal statistics. A number of authors [e. g., Field, 1987; Ruderman, 1994] have found that the marginal distributions of both image derivatives as well as wavelet coefficients are strongly non-Gaussian. Figure 4.2 shows the marginal log-histograms of the horizontal and vertical derivatives, as well as the gradient magnitude, all computed over the image database used here. As we can see, the marginal log-histograms have a much stronger peak than a Gaussian distribution with the same mean and variance. Also the tails of the distribution are very heavy; that is they fall off relatively slowly away from the mean. An indicator for that is the kurtosis of the distribution [Abramowitz and Stegun, 1964, § 26.1]

$$\kappa = \frac{E\left[(x-\mu)^4\right]}{E\left[(x-\mu)^2\right]^2}. \tag{4.1}$$

Figure 4.2 gives the kurtosis values for the derivative histograms; any value above 3 indicates "super-Gaussian" (or heavy-tailed) behavior. Many authors have attributed this phenomenon to object boundaries and the resulting discontinuities, which result in large jumps in intensity and cause substantial probability mass in the tails of the distribution

---

[2] We should note in general that most of the properties of natural images that we describe in this section only apply to large collections of natural images and not necessarily to single images.



(a) Log-histogram of responses of 8 random filters (see text).

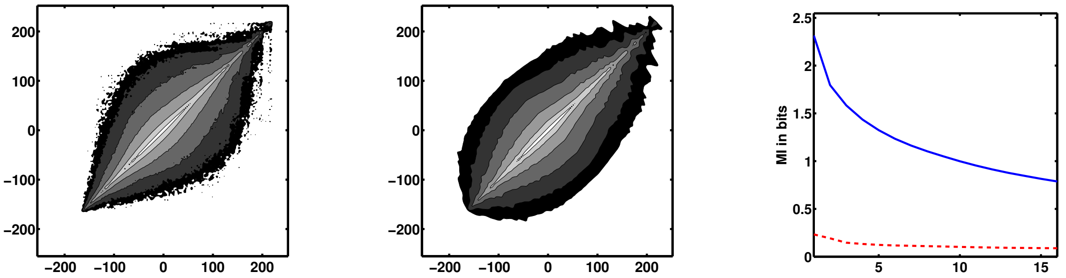
(b) Log-histogram of the horizontal derivative for 3 spatial scales (1, 2, 4).

Figure 4.3: **Other derivative histograms.**

[e. g., Field, 1987; Ruderman, 1994]. Lee et al. [2001] validated this interpretation by showing that a so-called "dead leaves" model of images [Matheron, 1968] gives rise to very similar statistical properties as natural images, including heavy-tailed derivative histograms. Such a dead leaves model is a very simple model of scene composition and occlusions, in which randomly sized shapes "fall down" onto the image plane and occlude one another. Huang [2000] showed that not only derivative and wavelet responses have such heavy-tailed properties, but that this is true even for random, zero-mean linear filters. Figure 4.3(a) shows the responses to 8 random zero-mean filters of size 7×7 pixels, where the filter coefficients were drawn from a unit normal distribution and the filter norm of each filter was set to 1 afterwards.

Scale invariance. Another interesting aspect of natural images is that their statistics are approximately scale invariant [e. g., Ruderman, 1997]. This means that, for example, derivative statistics are not strongly affected by the spatial resolution at which they are measured. We can, for example, subsample the images by block-averaging patches of $n \times n$ pixels and look at the derivative statistics of the subsampled images. Figure 4.3(b) shows derivative histograms for $n = 1, 2, 4$. As we can see, they look rather similar and are all heavy-tailed. Ruderman [1997] and others attributed this to the fact that objects in natural scene typically occur throughout a large range of sizes.

Joint statistics. Apart from marginal statistics, it is also revealing to study the joint statistics of two pixels or two wavelet coefficients [cf., Huang, 2000]. Figure 4.4(a) shows the joint log-histogram of two horizontally neighboring pixel values x_1 and x_2, which have been normalized by subtracting the global image mean. The "tilt" of the distribution, that is the fact that the main axis of variation is diagonal, suggests very strong statistical



(a) Joint log-histogram of pixels x_1 and x_2 (MI = 2.32bit).

(b) Approximation from marginals of $x_1 + x_2$ and $x_2 - x_1$.

(c) Mutual information as a function of pixel distance.

Figure 4.4: **Joint statistics of neighboring pixels.** (a) shows the joint log-histogram of pixels x_1 and x_2, and (b) displays an approximation from two marginals. (c) shows the mutual information in bits between pixel values at a given (horizontal) distance (solid blue). It also shows the mutual information between two "difference observations" at varying distances (dashed red); each such difference observation is the difference $x_2 - x_1$ of two neighboring pixel values.

dependence between the pixels. The mutual information [MacKay, 2003, § 8]

$$I(x_1; x_2) = \sum_{x_1, x_2} p(x_1, x_2) \log_2 \frac{p(x_1, x_2)}{p(x_1)p(x_2)} \tag{4.2}$$

between the two pixels x_1 and x_2 is 2.32 bits and thus quite large. Furthermore, the shape of the contour lines indicates non-Gaussian behavior: If the joint distribution was Gaussian, the log-histogram would have elliptical contour lines, but the histogram clearly does not. In Figure 4.4(b), we can see that the joint histogram can be relatively well approximated by taking the product of marginal distributions of $x_1 + x_2$ and $x_2 - x_1$ [cf., Huang, 2000]. This suggests that the mean and the difference of neighboring pixel values are largely independent. This provides another motivation for the discussion from Section 3.6: Dependent random variables in an image can be transformed using "difference observations", which makes them more independent.

Now we may ask what happens if we use such difference observations and model an image in terms of differences between neighboring pixels? This is precisely what most pairwise Markov random field models of images do (see Section 2.2). One important fact that we have not mentioned so far is that images have quite long-range correlations that go beyond direct neighbors. Figure 4.4(c) shows that the mutual information is large even for relatively distant pixels (solid blue line). If instead of the pixel values we use difference observations as just discussed and measure the mutual information between difference observations at various distances, we can see that the mutual information and thus the dependence is dramatically reduced (dashed red line). While this shows that difference observations are

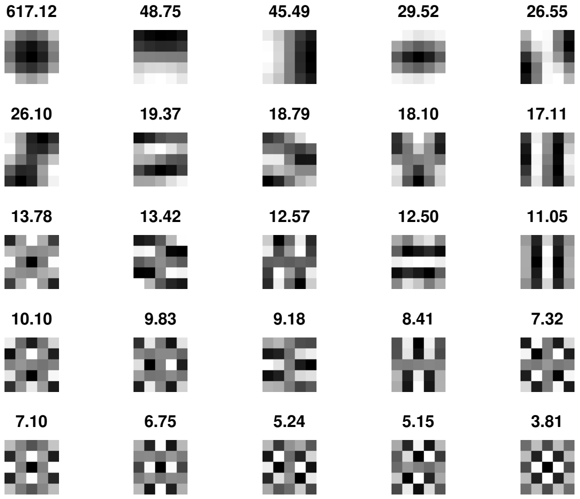


Figure 4.5: **Principal components of 5×5 natural image patches** along with their singular value (see text for more details). The components are scaled for display.

indeed quite helpful for modeling image data, since they alleviate long-range dependencies, there is still a non-negligible amount of long-range dependence that is not captured by these difference observations. Consequently, pairwise MRFs are not sufficient for modeling long-range correlations in natural images.

Principal component analysis. A final interesting aspect is the form of the principal components of natural images. In Figure 4.5 we show the 25 principal components of natural image patches of 5×5 pixels from our image database along with the singular values of the components. Note that we did not remove the mean from the data, hence the first principal component accounts for the mean of the data. Even though it appears to be smoothly varying due to the scaling employed for display purposes, it is in fact almost constant. The remaining principal components have a form that is very commonly encountered in spatial data, not only in images. In particular, the principal components are very similar to image derivatives of various orientations; the order of the derivatives increases as the singular value decreases.

4.1.2 Linear vs. logarithmic vs. gamma-compressed intensities

A number of studies on the statistics of natural images use logarithmic intensities [e. g., Huang, 2000]. In order to understand this further, let us first denote the image intensity, or more precisely the image irradiance, as $I_r(i,j)$. If we assume that it is purely caused by diffuse reflection, then the intensity is a combination of the diffuse reflectance $\rho(i,j)$ at the scene point corresponding to pixel (i,j) as well as the lighting $L(i,j)$, which encompasses both properties of the light source(s) itself as well as of the surface orientation [cf.,

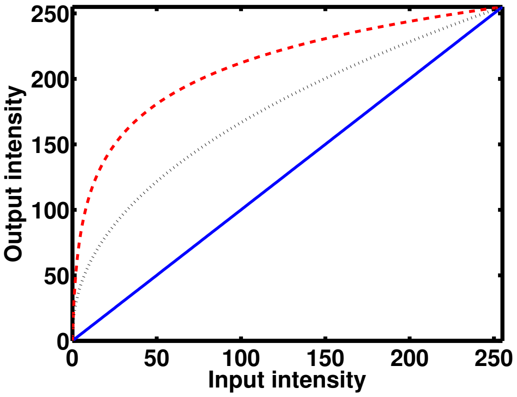


Figure 4.6: **Three image intensity transformations:** linear transformation (solid blue), logarithmic transformation (dashed red), and gamma compression (dotted black).

Foley et al., 1990, § 16.7]:

$$I_r(i,j) = \rho(i,j) \cdot L(i,j). \tag{4.3}$$

If we take the log of the intensity, the two contributions become additive:

$$\log I_r(i,j) = \log(\rho(i,j)) + \log(L(i,j)). \tag{4.4}$$

If the overall lighting level is now changed, for example, by making everything c times as bright, then the log-intensity is only changed by an additive constant. This seems like a desirable property, because we can make image models invariant to global illumination changes by making them invariant to the mean of the log-intensity data.

Alternatively, we could also consider gamma-compressed intensities. When image intensities are recorded by an image sensor, they are typically not stored directly (i. e., as linear intensity values), but instead they are gamma corrected, in particular gamma compressed [Poynton, 1998]:

$$I_g(i,j) = I_r(i,j)^{1/\gamma} - \epsilon. \tag{4.5}$$

The majority of commonly used image formats, including JPEG, store gamma-compressed images. We typically assume standard parameters for PCs of $\gamma = 2.2$ and $\epsilon = 0$. This correction has two purposes: (1) Historically, CRT screens have a response curve (gamma expansion) that corresponds to the inverse of this correction. So outputting a gamma-compressed image on a CRT results in the correct stimulus; (2) A side effect of the gamma compression is that the discrete gray levels have a perceptually uniform spacing [Poynton, 1998]. This means that the perceptual difference of neighboring gamma-compressed gray values is identical throughout the entire dynamic range of the screen. A logarithmic transform also increases perceptual uniformity, but not as well as a gamma correction.

It is now a somewhat philosophical question whether we want to model the statistics of log intensities, linear intensities, or gamma-compressed intensities. Figure 4.6 shows a comparison of the three intensity transformations. Since our goal in this chapter is to achieve optimal perceptual quality, we decided to use gamma-compressed intensities for the most part as they have the best perceptual uniformity. The image database we are using for our experiments consists of gamma-compressed intensities, which means that we work with the image data exactly as stored in the image files. For some of our later experiments, we will also require linear and log intensities. To obtain these, we undo the gamma compression and apply the respective transformation.

4.2 FoE Model of Natural Images

In the previous section we have seen a number of important statistical properties of natural images: They have very heavy-tailed derivative histograms, and even responses from random linear filters have very heavy-tailed responses. Secondly, natural images have scale invariant statistics. And finally, images show strong and rather long-range correlations, which can be reduced using difference observations, but not entirely removed. With these observations in mind, we can now model prior distributions of natural images using Fields of Experts, and then discuss how well the FoE captures the various statistical properties. Applying the FoE framework to modeling natural images will let us address two of these important properties of natural images. First, the heavy-tailed nature of random linear filter responses motivates us to choose heavy-tailed expert functions as will be discussed in more detail below. Second, the large cliques in the FoE model allow us to go beyond pairwise models, which we have just shown to be insufficient to model all long-range correlations.

Let us first return to the form of the experts. In a pairwise Markov random field, the shape of the potential functions is somewhat directly tied to the shape of the derivative marginals. Hence, there is a direct motivation for choosing heavy-tailed expert models, such as the Student t- or the Charbonnier expert from Section 3.2.1. If our graphical model $G = (V, E)$ for images was tree-structured, then we could rewrite the density under the graphical model in terms of pairwise and individual *marginals* as follows [Wainwright and Jordan, 2003, § 4.2.3]:

$$p(\mathbf{x}) = \prod_{s \in V} p(x_s) \prod_{(s,t) \in E} \frac{p(x_s, x_t)}{p(x_s)p(x_t)}. \tag{4.6}$$

This means that if the single node marginals are uniformly distributed (which they roughly are in natural images), then the optimal *potentials* for this tree-structured model are the pairwise *marginals* itself. So if we observe that the difference $x_s - x_t$ of neighboring pixels has a certain heavy-tailed distribution (on the data), then a good potential for the model should have the same heavy-tailed form. However, the graphical models that we consider here are not tree-structured, and Eq. (4.6) no longer holds in general. Nevertheless, it can be shown [Wainwright and Jordan, 2003, § 6.2.2] that even for loopy graphs, combining pairwise and individual marginals as in Eq. (4.6) leads to a fixed point of the Bethe free energy, which is related to the loopy belief propagation algorithm (see Sections 2.1.1 and 3.4). If loopy BP converges, it finds a fixed point of the Bethe free energy. This means that even in the loopy case, using the marginals of the data directly as potential functions is "optimal" (i. e., locally optimal in terms of the Bethe free energy). Despite that, this property of pairwise MRF models has, with few exceptions [Sebastiani and Godtliebsen, 1997; Freeman et al., 2000; Tappen et al., 2003; Ross and Kaelbling, 2005], not been exploited widely.

In the high-order MRF case that we consider here, there does not seem to be such a direct connection between the marginals and the optimal potentials. Hence we (somewhat arbitrarily) choose some heavy-tailed potential function for use in the FoE framework and later study the impact. For all applications in Sections 4.3 and 4.4 we use the Student t-experts from Eq. (3.10). The FoE model of natural images is thus written as

$$p_{\mathrm{FoE}}(\mathbf{x}; \Theta) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K} \prod_{i=1}^{N} \left(1 + \frac{1}{2}(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)})^2\right)^{-\alpha_i}. \tag{4.7}$$

Later in Section 4.5, we will also explore the use of Charbonnier experts.

For modeling natural images, we typically use FoEs with cliques of 5 × 5 pixels. We mostly use 24 filters, which means that these is one filter for each degree of freedom in each clique patch (not considering the patch mean, which is not modeled). Another model we often use is based on 3 × 3 cliques and 8 filters. We train each model on 20000 images of size 15 × 15, which were randomly cropped from the image database described in the previous section. As described in Section 3.3.4, the filters are defined in a coordinate system that encourages high-frequency components, since that improves the application performance (see Section 4.5.1 for more details). Learning is done using contrastive divergence as described in detail in Section 3.3. Figure 3.4 in the previous chapter shows the filters and the corresponding expert parameters that were learned for a 5 × 5 model with 24 filters.

Applications of the model. Such a prior model of natural images has a variety of applications, which include image denoising and image inpainting. Both will be discussed in the subsequent sections. Beyond that, the model is, for example, applicable to the problems of super-resolution and demosaicing [see e. g., Tappen et al., 2003]. There, the FoE model would help to find a natural looking full-resolution image that agrees with the subsampled input image. As far as we are aware, there has not been a thorough analysis of FoEs in this context[3]. Other applications include blind and non-blind deconvolution [Fergus et al., 2006], image-based rendering, and new-view synthesis. Woodford et al. [2006] applied an FoE model of natural images to image-based rendering, and obtained encouraging results while enabling faster inference than was possible with example-based methods [Fitzgibbon et al., 2003]. Sun and Cham [2007] showed that FoEs can be used to aid removing compression artifacts in highly compressed JPEG images.

The image models that we use in this chapter are all based on gray level images. As shown in Section 4.3, such a model can nevertheless be applied to color images by simply applying it independently to the channels of a suitable color space. It is also quite easily possibly to actually extend the model to color images. To that end we could simply use cliques (or factors) of size $m \times m \times 3$ (for 3 color channels); the filter masks then apply to all 3 color channels simultaneously. Multi-spectral data, for example from satellite imagery, could be handled in a similar way. McAuley et al. [2006] followed this approach and reported encouraging results for color denoising with an FoE model of color images[4].

Another potentially interesting application of FoE image models is to other image data, for example from medical imaging (MRI, CT, and ultrasound), or radar data. Unfortunately, these types of data are usually too noisy to be used for training, as we would learn a model that incorporates noise. Developing methods for training FoE models from noisy data, for example using an EM-type algorithm [Neal and Hinton, 1998], where the noise-free images are inferred at each step based on a physical model of the noise, appears to be an interesting avenue for future work.

4.2.1 Interpreting the model

It is relatively difficult to develop an intuitive understanding of how the FoE model works, particularly also regarding the nature of the linear filters that are learned from data. As we saw in Figure 3.4, the filters learned for an FoE model of natural images look quite unlike the usual kinds of filters that have been used in image processing, such as smooth derivative (Gabor) filters. In particular, they have very high frequency and sometimes localized structure. We make this more explicit here by looking at the frequency response of some of the filters. Figure 4.7 shows frequency response plots of some of the most relevant filters. We can see that the filters typically have their largest responses for high spatial frequencies in either vertical, or horizontal direction, or both. Nevertheless, as we

---

[3] Some of our preliminary experiments (not documented here) indicate that the local optimization methods we use for inference in this chapter may not be very suitable for problems where high-frequency structure (with a resolution that is higher than the input resolution) needs to be obtained. The development of more advanced inference techniques would likely benefit these applications directly.

[4] Note, however, that they did not fully train the model.

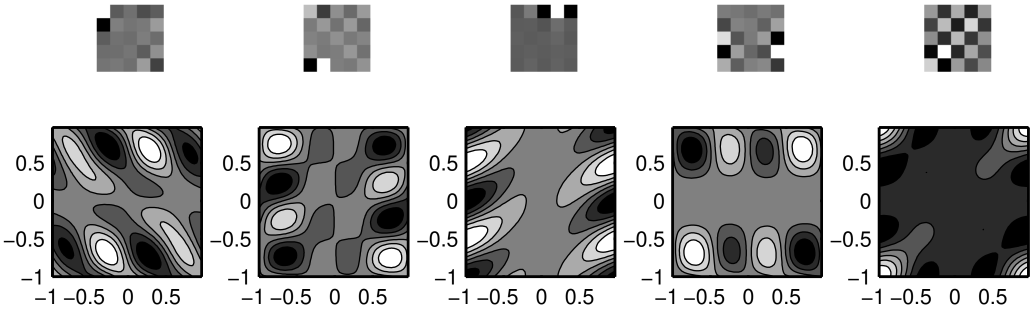


Figure 4.7: **Frequency responses of FoE filters.** The frequency responses of the five filters with the largest α_i are shown below the respective filter. The axes denote the spatial frequencies, where 0 represents a constant image and ±1 represents high spatial frequencies. White indicates a strong frequency response, black a weak response.

will see in Section 4.5, these unusual, high-frequency filters are not an artifact of the learning algorithm, but do indeed improve the application performance over more standard filters such as those found by patch-based models.

To get a better idea of what the filters do, let us look at their response to a particular natural image: Figure 4.8 shows the filter responses of all 24 filters on a particular natural image, which is displayed as well. Each filter response is shown along with the filter that gave rise to the response. We should note that these are the same filters as shown in Figure 3.4. When looking at the filter responses, it is important to keep in mind that in the model each individual filter response is never considered without considering all other filter responses at the same time. In other terms, the ensemble of filters and experts is what defines the FoE model, and not necessarily what an individual filter and expert does. We can see from the responses that, despite their unusual nature, the filters respond to intuitive image features such as edges and textures at various orientations and even scales, but they generally respond only quite infrequently. This can be attributed to the heavy-tailed expert function, which strongly prefers filter responses close to zero, and dislikes large responses. This means that it is quite possible that the filters match structures that only very rarely occur in natural images. Hinton and Teh [2001] noted that a PoE model with Student t-experts (or some other heavy-tailed expert) is very suitable for modeling constraints that are "frequently approximately satisfied". They conclude that the linear features found should be best at ignoring things that are typical in natural images. Hence, finding filters that only rarely respond to natural image structures is encouraged by the FoE model, which in turn helps us understand why the learned filters may still be sensible despite looking unusual (see also [Weiss and Freeman, 2007b]).

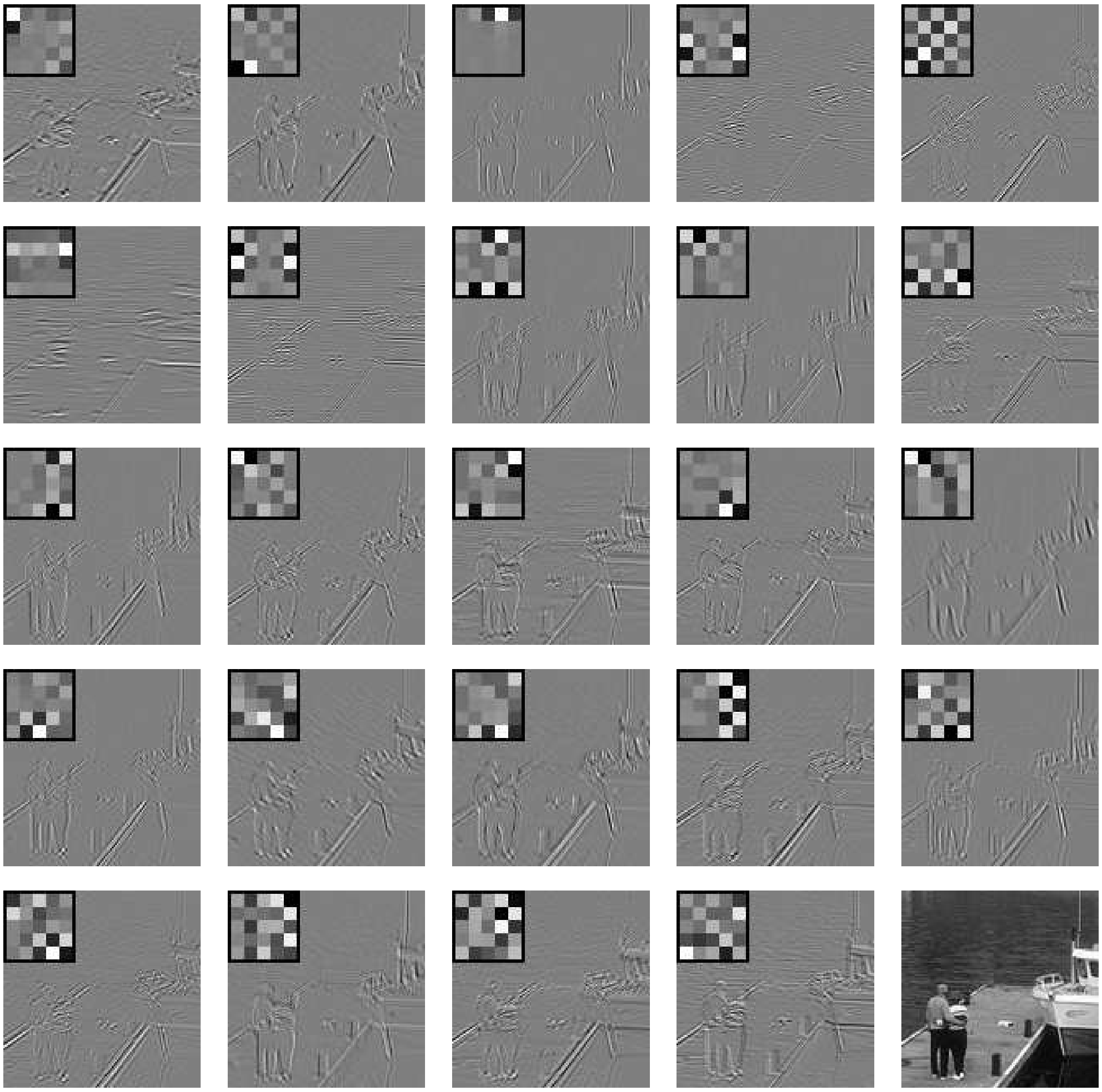
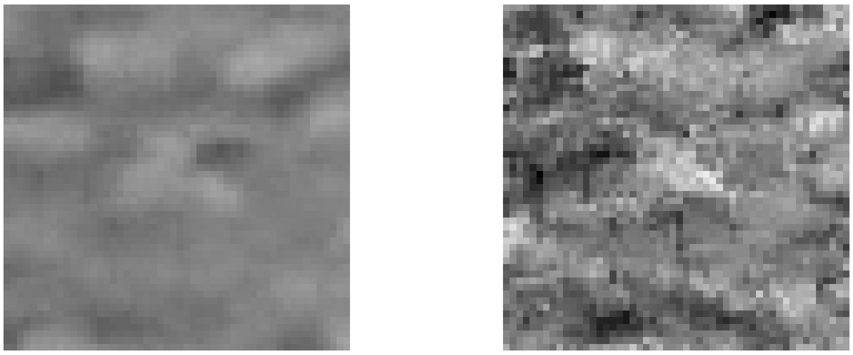


Figure 4.8: **Filter responses of the FoE filters.** Each part shows the filter response to the image in the bottom right along with the filter that gave rise to the response.



(a) Sample from the original model.

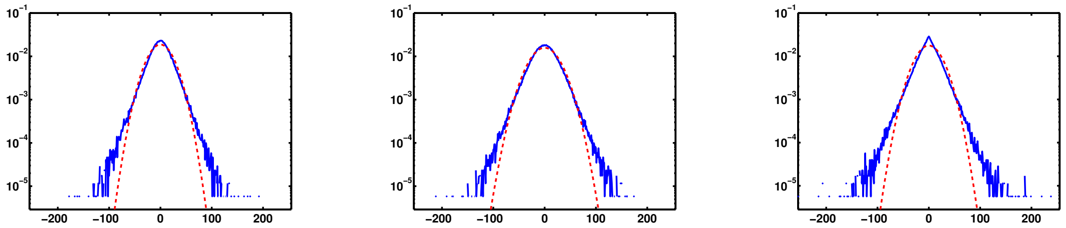
(b) Sample from a model with scaled expert parameters (see text).

Figure 4.9: **Samples from a 3×3 FoE model with 8 filters.**

4.2.2 Other aspects of the model

Before moving on to applications of the FoE model of natural images, it will be interesting to look at a few other aspects of the model, in particular as to how well FoEs model the various characteristics of natural images that we discussed in Section 4.1. One important motivation for the FoE was that we wanted to better capture complex long-range correlations than is possible with pairwise MRF models. The experiments in the subsequent chapters will illustrate how the FoE improves on pairwise models in this respect, hence we will not discuss this in more detail here.

One advantage of generative models is that they allow us to sample from the model, which we can use to check how much the samples look like our data, in this case like natural images. Since for illustration we want to sample a rather large image for which hybrid Monte Carlo sampling is not particularly effective, we instead use Gibbs sampling, where each pixel is iteratively sampled by conditioning on all other pixels. Sampling is done in a random order using all values in $\{0, \ldots, 255\}$. Figure 4.9(a) shows a sample from a 3×3 FoE model with 8 filters (chosen to reduce the computational effort). As we can see the sample looks very smooth and relatively unlike natural images. In the application to image denoising in the following section, we find that FoE models benefit from a "weight" ω that reduces the smoothing effect of the prior. This weight is equivalent to scaling down all the α_i with a constant (e. g., $\hat{\alpha}_i := \omega \cdot \alpha_i$). In denoising with the 3×3 model used here, we find that $\omega \approx 0.5$ works well. If we sample from a model where the expert parameters are scaled down according to this weight, then we obtain more "realistic" looking samples. Figure 4.9(b) shows such a sample, which exhibits both smooth areas, but also discontinuities. Nevertheless, it is worth noting that we would never expect a sample to really look like an actual natural image; we can only expect the samples to exhibit some



(a) Marginal of horizontal derivative.

(b) Marginal of vertical derivative.

(c) Marginal of a learned FoE filter.

Figure 4.10: **Marginal statistics of samples from the FoE model.** Each plot shows a marginal log-histogram computed from model samples, as well as a fit with a Gaussian (red, dashed).

of the structure of natural images. The necessity of the weight ω suggests that there is a possible bias in the contrastive divergence learning procedure, which causes these incorrect expert parameters. Nonetheless, using more contrastive divergence steps per iteration (and thus making CD more similar to maximum likelihood estimation) or using a Gibbs sampler instead of the usual hybrid Monte Carlo sampler does not remove the need for the weight in denoising (see Section 4.5.10 for details).

Another interesting aspect is to see how well the FoE captures marginal statistics of natural images. We can use samples from the model to compute marginal derivative statistics, as well as filter response statistics of the learned filters. Given the smooth nature of the model samples as just discussed, the response statistics of the unmodified model are very close to Gaussian. If we apply the same weight ω to the model before sampling and computing marginal histograms, we instead get responses that are more heavy-tailed. Figure 4.10 shows horizontal and vertical derivative statistics, as well as the filter response statistics of one of the FoE filters. Each plot also shows a fit of a Gaussian distribution to the response. As we can see, the response does have heavier tails than a Gaussian, but the peak is not nearly as pronounced and the tails are not nearly as heavy as is the case of natural images (cf. Figs. 4.2 and 4.3(a)). The response to the learned filter shows a more clearly pronounced peak, but the overall shape is still less heavy-tailed than a Student t-distribution. From our discussions about maximum entropy models in Section 3.5.2, this may not come as a huge surprise, since Fields of Experts do not constrain marginal filter responses to match those of the data. This is quite interesting from two points of view: First, pairwise MRF models capture marginal derivative statistics very well (essentially by construction), yet do not work nearly as well as the FoE in real applications, as we will see later. Secondly, the FRAME model, which is a maximum entropy model, preserves marginal filter statistics by construction, but also does not lead to very natural looking image restoration results. Samples from a FRAME model of natural images [Zhu and Mumford, 1997] have very different qualities than the ones from the FoE model. In particular, they are piecewise constant, which is reflected in piecewise constant and relatively unnatural-looking image restoration results.

This opens up two questions that will not be answered in this dissertation: (1) What are the important statistical properties of natural images *beyond* their marginal statistics? Modeling marginal statistics may be important, but the findings here suggest that it is possible that modeling complex long-range correlations well is even more important; (2) How can we improve the FoE model so that it captures the marginal statistics as well? In particular, can we choose better expert models? The fact that the marginal filter response statistics are not as heavy-tailed as we would like them to be may indicate that the tails of the already very heavy-tailed Student t-experts are *not heavy enough*. Yet, using even more heavy-tailed experts is not without problems. Sallee and Olshausen [2003] use a mixture of a Dirac delta and a broad Gaussian to define very heavy-tailed "experts" in their model (even though they are not necessarily thought of in this way). But they require Gibbs sampling for learning and inference. Such an expert is not directly usable in the FoE framework, because it is not differentiable. But it would be very interesting to study if similar differentiable experts can be found, or if it is possible to generalize the learning procedure developed here, while still avoiding the computational difficulties associated with the model by Sallee and Olshausen.

So far, we have not discussed one other important aspect of the statistics of natural images: their scale invariance. The FoE framework as presented only models images at the finest spatial scale. In particular, its (say) 5×5 filters are too small to capture statistics at very coarse spatial scales, but computational considerations prevent us from making them much bigger. The FRAME model [Zhu and Mumford, 1997], other the other hand, uses derivative filters of various spatial scales to capture marginal statistics at various spatial scales. One important way in which Fields of Experts can be developed further in the future is by extending the formulation with the goal of modeling scale invariant statistics. In particular, such an extended model could allow for filters to be applied at several spatial scales.

4.3 Application: Image Denoising

Noise is ubiquitous in digital images. In most cases, it comes directly from the image sensor itself or from the subsequent processing within the camera. In CCD cameras, for example, there is shot noise from the stochastic nature of the photons arriving at the sensor element, noise from the analog processing inside the sensor, noise from quantization, and from a few other sources (see Healy and Kondepudy [1994] and Tsin et al. [2001] for detailed overviews of the various sources of CCD sensor noise). Other sensors (CMOS, etc.) and images sources have different noise sources and characteristics. Here, we will not carefully study the characteristics of any particular noise model, but instead assume a very simple noise model. As is very common in the denoising literature, our experiments assume that the true image $\mathbf{x}$ has been corrupted by additive, i. i. d. Gaussian noise with zero mean and known standard deviation σ:

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}). \tag{4.8}$$

It is nevertheless important to note that the techniques presented here can easily be generalized to other noise models, as long as the noise distribution is known and its logarithm is differentiable. This is in contrast to a large part of the denoising literature that is solely based on having Gaussian noise. Also, in practice the amount of noise in the image may not necessarily be known. The extension of our exposition to this case of "blind" denoising, for example using robust data terms, automatic stopping criteria, or using an automatic noise level detection mechanism [Liu et al., 2006] will remain the subject of future work.

4.3.1 Review of the denoising literature

In image denoising it is necessary to assume some form of prior knowledge about natural images; otherwise, it is impossible to separate the signal from the noise. Denoising techniques can be roughly divided into techniques that are based on probabilistic models of prior knowledge, and those that are based on other forms of prior knowledge. We will now review the most important methods from both categories.

Denoising using non-probabilistic methods in the image domain. Many natural images have parts where they are locally smooth. Hence, the most basic prior assumption used about natural images is that of local smoothness. In its very basic form this is exploited by linear filtering techniques using Gaussian filter kernels. These simple filtering techniques can also be interpreted in terms of partial differential equations (PDEs) modeling heat transfer, and are also known as linear diffusion methods. Because of the global smoothness assumptions, these very simple filters also smooth over discontinuities that exist in images. A large class of methods was developed to overcome this problem by assuming that images are only piecewise smooth. Perona and Malik [1990] proposed the anisotropic diffusion method for denoising, which is more precisely characterized as nonlinear diffusion. The win over linear diffusion methods comes from the fact that the amount of smoothing varies with the local structure of the image; less smoothing is performed at boundaries than in smoothly varying areas. Starting from this work, there has been a large body of literature of nonlinear diffusion methods that overcomes some problems in the initial approach [Catté et al., 1992; Weickert, 1996; Charbonnier et al., 1997]. Anisotropic extensions adapt the smoothing also to the direction of the local gradient structure [Weickert, 1996; Black et al., 1998; Scharr and Weickert, 2000]. Other recent extensions include complex-valued diffusion processes [Gilboa et al., 2004], which encourage smooth gray value ramps. An overview over this large body of literature is, for example, provided by Weickert [2001]. Even though these techniques are typically not directly motivated by statistical properties of natural images, such connections can be made [Black et al., 1998; Black and Sapiro, 1999; Scharr et al., 2003].

While these PDE methods are typically motivated from the point of view of diffusion processes, a number of them have direct connections to variational methods, where denoising is performed by minimizing a global energy functional [Rudin et al., 1992; Schnörr, 1994; Schnörr et al., 1996]. Variational methods are discussed in more detail in Section 2.2.1. Both diffusion and typical variational methods have a number of important mathematical properties, but in comparison to many other techniques mentioned below, their performance is mostly lacking today.

Another category of techniques are neighborhood filters, which modify a pixel's intensity based on a weighted combination of its neighborhood. The particular form of prior knowledge exploited there is that images are often locally self-similar. Prominent examples are the bilateral filter [Tomasi and Manduchi, 1998], non-local means [Buades et al., 2004, 2005], and related methods [Polzehl and Spokoiny, 2000; Kervrann and Boulanger, 2006]. These approaches are currently among the most accurate denoising methods. While these approaches are not based on probabilistic models and are not directly motivated by the statistics of natural images, there are statistical connections that admit the study of convergence to the true, noise free image [Buades et al., 2004].

Denoising using probabilistic methods in the image domain. Among probabilistic methods for denoising in the image domain, approaches based on Markov random field models have been the most prominent. Their application to denoising dates back at least to the works of Kashyap and Chellappa [1981], Geman and Geman [1984], and Besag [1986], and continues into the more recent literature [Sebastiani and Godtliebsen, 1997; Felzenszwalb and Huttenlocher, 2004]. Gaussian MRFs, including models with high-order cliques, have also found substantial use, but have the drawback that they often strongly smooth edges [Kashyap and Chellappa, 1981; Tsuzurugi and Okada, 2002]. Non-Gaussian high-order models have only been used quite rarely, with the exception of the FRAME model [Zhu and Mumford, 1997]. While quite different from traditional MRF methods, the recent nonparametric UINTA method [Awate and Whitaker, 2006] is also based on Markov assumptions. While MRFs have been popular for denoising in the past, their performance has recently fallen short of other methods, particularly to those that operate in the wavelet domain.

Denoising in transformed domains. Currently, some of the most accurate denoising methods in the literature fall within the category of wavelet methods, in which the image is: (1) decomposed using a large set of wavelets at different orientations and scales (often an overcomplete wavelet basis); (2) the wavelet coefficients are modified based on some criterion (often "shrinkage" or "coring"); and (3) the image is reconstructed by inverting the wavelet transform. In the second step, the wavelet coefficients are typically shrunk toward zero, for example using a fixed threshold that only depends on the amount of noise in the images [Donoho, 1995]. Recent methods model the fact that the marginal statistics of the wavelet coefficients are non-Gaussian, and use a so-called Bayesian threshold [Simoncelli, 1999]. More refined methods also take into account that neighboring coefficients in space or scale are not independent [Portilla and Simoncelli, 2000; Portilla et al., 2003; Gehler and Welling, 2006]. The most accurate of these methods [Portilla et al., 2003] models the these dependencies using a Gaussian scale mixture and uses a Bayesian decoding algorithm. Portilla et al. use a pre-determined set of filters and hand select a few neighboring coefficients (e. g., across adjacent scales) that intuition and empirical evidence suggest are statistically dependent. Their work also contains an excellent review and quantitative evaluation of the state of the art in wavelet denoising.

While some of these methods are based on probabilistic models, it is important to understand that they model the prior density of wavelet coefficients, and not of natural images. Because overcomplete wavelet bases are usually employed, these wavelet approaches do not directly correspond to a prior model of images.

Other image transformations have been studied as well, including curvelets [Starck et al., 2002], which overcome some of the problems of wavelet transformations and promise reduced artifacts. Furthermore, interesting relationships between denoising methods in the wavelet domain and other methods have been developed: For example, wavelet shrinkage was shown to be related to diffusion methods [Steidl et al., 2004]. Other recent work has explored the connection between shrinkage in overcomplete wavelet domains and shrinkage procedures in the image domain [Elad et al., 2006a,b]. Furthermore, a number of interesting methods have been developed that find suitable image transformations while performing denoising [Elad and Aharon, 2006; Donoho et al., 2006]. One interesting aspect of the work by Elad and colleagues is that it allows one to define a prior model for images, and can thus be used for other applications such as inpainting [Mairal et al., 2006].

Denoising using patch-based models. A final, somewhat smaller category of denoising methods are patch-based denoising algorithms. They often exploit statistical prior knowledge on patches and combine denoised patches in some way to form a coherent denoised image. Such methods are for example based on Products of Experts [Welling et al., 2003] or independent component analysis [Hyvärinen et al., 2003].

4.3.2 Denoising using Fields of Experts

In contrast to a number of the above schemes, we focus on a Bayesian formulation with a probabilistic prior model of the spatial properties of images. As in a general Bayesian image restoration framework, our goal is to find the true image $\mathbf{x}$ given an observed image $\mathbf{y}$ by maximizing the posterior probability $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})$. Because of our Gaussian noise assumption (cf. Eq. (4.8)), we write the likelihood as

$$p(\mathbf{y}|\mathbf{x}) \propto \prod_{j=1}^{M} \exp\left(-\frac{1}{2\sigma^2}(y_j - x_j)^2\right), \tag{4.9}$$

where j ranges over the pixels in the image. Furthermore, we use the FoE model as the prior, i. e., $p(\mathbf{x}) = p_{\mathrm{FoE}}(\mathbf{x})$.

As discussed in Section 3.4 maximizing the posterior probability of such a graphical model is generally hard. In order to emphasize the practicality of the proposed model, we refrained from using expensive inference techniques and use the continuous optimization techniques introduced in Section 3.4. In particular, we performed a gradient ascent on the logarithm of the posterior probability as in Eq. (3.44). The gradient of the log-likelihood is written as

$$\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) = \frac{1}{\sigma^2}(\mathbf{y} - \mathbf{x}), \tag{4.10}$$

and is combined with the gradient of the log-prior from Eq. (3.24). By introducing an iteration index t, an update rate τ, and an optional weight ω for the prior, we can write a simple gradient ascent denoising algorithm as follows:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \tau \left[\omega \cdot \sum_{i=1}^{N} \mathbf{J}_i^{-} * \boldsymbol{\psi}'(\mathbf{J}_i * \mathbf{x}^{(t)}; \boldsymbol{\alpha}_i) + \frac{1}{\sigma^2}(\mathbf{y} - \mathbf{x}^{(t)})\right]. \tag{4.11}$$

In our experiments, we did not use this gradient ascent procedure directly, but used a more efficient conjugate gradient method based on the implementation of Rasmussen [2006]. Because this conjugate gradient technique is based on line-searches, it is not necessary to choose the update rate τ by hand. The optional weight ω can be used to adjust the strength of the prior compared to the likelihood. If both prior and likelihood were very accurately modeled and if we could find the global optimum of the denoising objective, such a weight would not be necessary. In practice, it can substantially improve performance, however. To make the interpretation easier, we parametrized this weight as $\omega(\lambda) = \frac{\lambda}{1-\lambda}$, where $\lambda \in (0, 1)$.

As observed by Zhu and Mumford [1997], such gradient ascent procedures are directly related to nonlinear diffusion methods. If we had only two filters (x- and y-derivative filters) then Eq. (4.11) is similar to standard nonlinear diffusion filtering with a data term. Also note that $-\psi_i = -\log \phi_i$ is a standard robust error function when ϕ_i has heavy tails, and that ψ_i is proportional to its influence function. This connects it to robust diffusion methods [Black et al., 1998].

Even though denoising proceeds in very similar ways to nonlinear diffusion, our prior model uses many more filters. The key advantage of the FoE model over standard diffusion techniques is that it tells us how to build richer prior models that combine more filters over larger neighborhoods in a principled way.

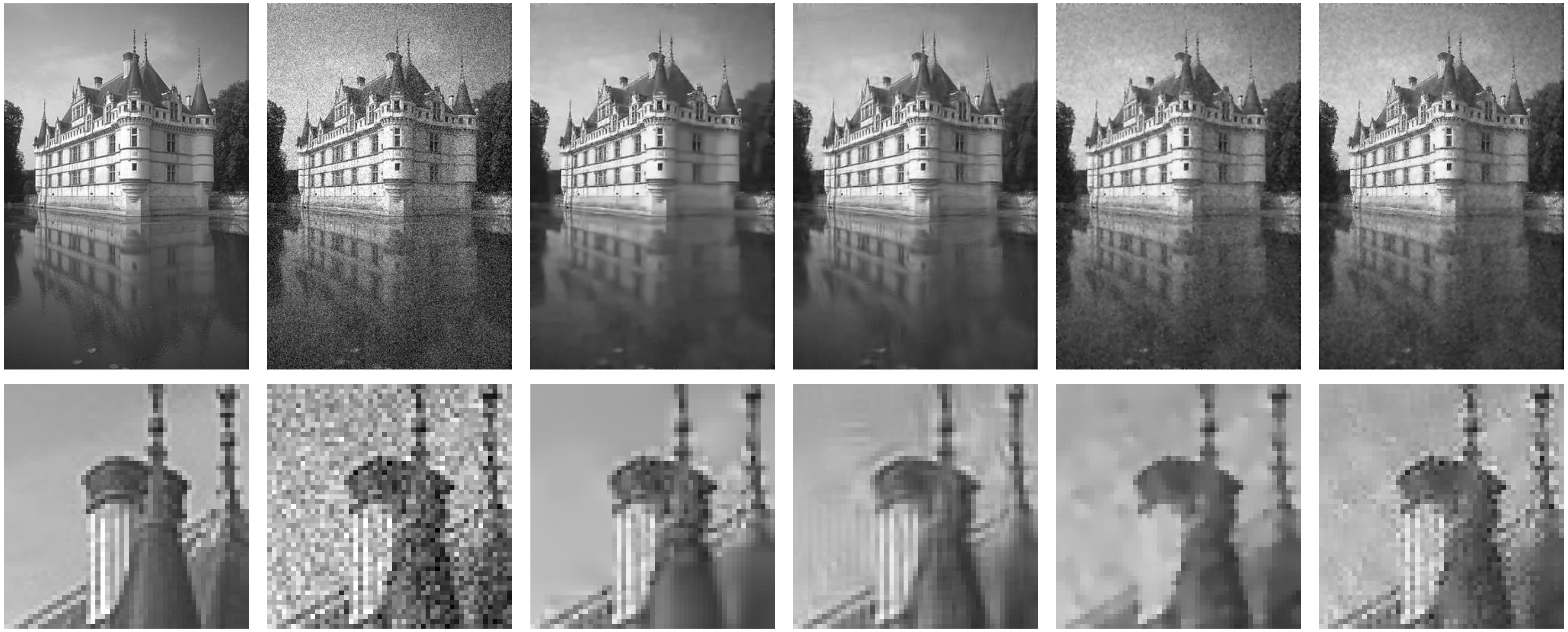
4.3.3 Experiments

Using the FoE model trained as in the previous section (5×5 cliques with 24 filters) we performed a number of denoising experiments. The evaluation of the denoising performance relied on two measurements: (1) The peak signal-to-noise ratio (PSNR)

$$\text{PSNR} = 20 \log_{10} \frac{255}{\sigma_e}, \tag{4.12}$$

where σ_e is the standard deviation of the pixelwise image error. PSNR is given in decibels (dB); a reduction of the noise by a factor of 2 leads to a PSNR increase of about 6dB. (2) The structural similarity index (SSIM) by Wang et al. [2004]. The PSNR is a very widely used evaluation criterion for denoising, but has the limitation that it does not fully reflect the perceptual quality of an image to the human observer. But in image denoising, our goal (usually) is to optimize perceptual performance. The SSIM provides a perceptually more plausible image error measure, which has been verified in psychophysical experiments. SSIM values range between 0 and 1, where 1 is a perfect restoration.

We performed denoising using at most 5000 iterations of conjugate gradient. In essentially all of the cases, the ascent terminated in fewer than 5000 iterations because a local optimum had been reached. Experimentally, we found that the best results are obtained with an additional weight as introduced above, which furthermore depends on the amount of noise added. We determined the appropriate λ trade-off parameter for denoising using an automatic training procedure that was carried out for each noise standard deviation that we wished to use. We manually picked a representative set of 10 images from the training database, cropped them randomly to 200×200 pixels, and added synthetic Gaussian noise



(a) (b) (c) (d) (e) (f)

Figure 4.11: **Denoising with a Field of Experts.** Full image (top) and detail (bottom). (a) Original noiseless image. (b) Image with additive Gaussian noise ($\sigma = 25$); PSNR = 20.29dB. (c) Image denoised using a Field of Experts; PSNR = 28.72dB. (d) Image denoised using the approach of Portilla et al. [2003]; PSNR = 28.90dB. (e) Image denoised using standard nonlinear diffusion (comparable to pairwise MRF); PSNR = 27.18dB. (f) Image denoised using Wiener filtering (with a 5×5 window); PSNR = 26.89dB.

(a) Lena (b) Barbara (c) Boats

(d) House (e) Peppers

Figure 4.12: **Commonly used images for evaluating denoising algorithms.**

of the appropriate standard deviation. Each artificially corrupted image was then denoised using the conjugate gradient method, and the optimal λ parameter with respect to the PSNR was determined in a two stage process: First, we denoised the training set using all $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. We then fit a cubic spline through the PSNR values for these λ values and found the value $\hat{\lambda}$ that maximized the PSNR. In the second stage, the search was refined to $\lambda \in \hat{\lambda} + \{-0.06, -0.04, -0.02, 0, 0.02, 0.04, 0.06\}$. The PSNR values for all λ values were again fit with a cubic spline, and the value λ^* that maximized the PSNR across all 10 training images was chosen.

Results were obtained for two sets of images. The first set consisted of images commonly used in denoising experiments (see Figure 4.12). The images were obtained from [Portilla, 2006a]. Table 4.1 provides PSNR and SSIM values for this set and various levels of additive Gaussian noise [cf. Portilla et al., 2003]. Portilla et al. [2003] report some of the most accurate results on these test images and their method is tuned to perform well on this dataset. We obtained signal-to-noise ratios that are close to their results (mostly within 0.5dB), and in some cases even surpassed their results (by about 0.3dB). To the best of our knowledge, no other Markov random field approach has so far been able to closely compete with such wavelet-based methods on this dataset. Also note that the prior was not trained on, or tuned to these examples. Our expectation is that the use of better MAP estimation

Table 4.1: **Peak signal-to-noise ratio (PSNR)** and **structural similarity index (SSIM)** for images (from [Portilla, 2006a]) denoised with FoE prior.

(a) PSNR in dB

σ	Noisy	Lena	Barbara	Boats	House	Peppers
1	48.13	47.84	47.86	47.69	48.32	47.81
2	42.11	42.92	42.92	42.28	44.01	42.96
5	34.15	38.12	37.19	36.27	38.23	37.63
10	28.13	35.04	32.83	33.05	35.06	34.28
15	24.61	33.27	30.22	31.22	33.48	32.03
20	22.11	31.92	28.32	29.85	32.17	30.58
25	20.17	30.82	27.04	28.72	31.11	29.20
50	14.15	26.49	23.15	24.53	26.74	24.52
75	10.63	24.13	21.36	22.48	24.13	21.68
100	8.130	21.87	19.77	20.80	21.66	19.60

(b) SSIM [Wang et al., 2004]

σ	Noisy	Lena	Barbara	Boats	House	Peppers
1	0.993	0.991	0.994	0.994	0.993	0.993
2	0.974	0.974	0.983	0.978	0.982	0.981
5	0.867	0.936	0.957	0.915	0.930	0.950
10	0.661	0.898	0.918	0.860	0.880	0.923
15	0.509	0.876	0.884	0.825	0.866	0.901
20	0.405	0.854	0.841	0.788	0.850	0.879
25	0.332	0.834	0.805	0.754	0.836	0.853
50	0.159	0.741	0.622	0.614	0.763	0.735
75	0.096	0.678	0.536	0.537	0.692	0.648
100	0.066	0.615	0.471	0.473	0.622	0.568

techniques will improve these results further.

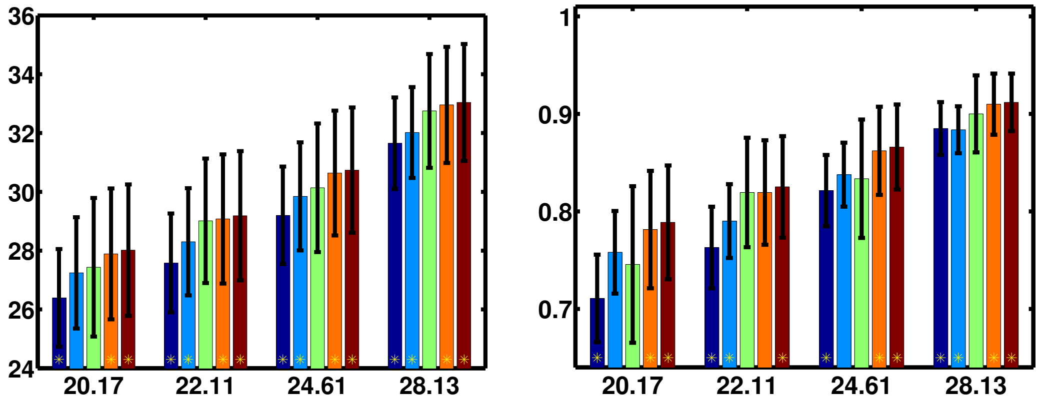
To test more varied and realistic images we denoised a second test set consisting of 68 images from the separate test section of the Berkeley segmentation dataset [Martin et al., 2001]. Figure 4.14(a) shows example images from this test set. For various noise levels we denoised the images using the FoE model, the method from [Portilla et al., 2003] (using the software and default settings provided by [Portilla, 2006b]), simple Wiener filtering (using MATLAB's `wiener2` with a 5 × 5 window), and a standard nonlinear diffusion scheme [Weickert, 1997] with a data term. This last method employed a robust Huber function and can be viewed as an MRF model using only first derivative filters. For this standard nonlinear diffusion scheme, a λ weight for the prior term was trained as in the FoE case and the stopping time was selected to produce the optimal denoising result (in terms of PSNR). Note that in case of the FoE denoising was *not* stopped at the point of optimal PSNR, but rather at convergence. Figure 4.11 shows the performance of each of these methods for one of the test images. Visually and quantitatively, the FoE model outperformed both Wiener filtering and nonlinear diffusion and nearly matched the performance of the specialized wavelet denoising technique. FoE denoising results for other images from this set are shown in Figure 4.13.

Figure 4.14 shows a performance comparison of the mentioned denoising techniques over all 68 images from the test set at various noise levels. The FoE model consistently outperformed both Wiener filtering and standard nonlinear diffusion in terms of PSNR, while closely matching the performance of the current state of the art in image denoising [Portilla et al., 2003]. A signed rank test showed that the performance differences between the FoE and the other methods were mostly statistically significant at a 95% confidence level (indicated by an asterisk on the respective bar). In terms of SSIM, the relative performance was very similar to that measured using the PSNR, with two notable exceptions: (1) When looking at the SSIM, the FoE performed slightly worse than nonlinear diffusion for two of the four noise levels, but the performance difference was not statistically significant in these cases. In the two cases where the FoE outperformed standard diffusion, the difference was significant, on the other hand. We should also keep in mind that nonlinear diffusion was helped substantially by the fact that it was stopped at the optimal PSNR, which is not possible in real applications. (2) On one of the noise levels, the FoE performed on par with the method of Portilla et al. [2003] (i. e., there was no significant performance difference). Overall, this means that in the majority of the cases the FoE performed significantly better than Wiener filtering and nonlinear diffusion, but also that the Wavelet method was still significantly better than the FoE (at a 95% confidence level).

As alluded to earlier, it is also possible to perform color denoising using this method. In particular, we transformed color images into YCbCr space in order to separate the luminance (Y) from the color contributions. We then denoised each channel independently using the procedure described above. While this is likely suboptimal compared to denoising with a prior model of color images, the performance was nevertheless visually pleasing. Figure 4.15 gives color results for the color version of the image in Fig. 4.11. We also show a comparison to nonlinear diffusion (again stopped at the optimal PSNR) and the method of Portilla et al. [2003], both of which were applied to color channels individually. The wavelet approach gave rather poor results when used in YCbCr space; denoising in RGB space gave good results, however. McAuley et al. [2006] model color images using Fields of Experts and found that a color model improves performance over processing the channels separately. The reader is referred to this paper for more results on color image denoising.

Figure 4.13: **Other Field of Experts denoising results.** Noisy input (left) and denoised image (right).

(a) Subset of the images used to evaluate the FoE model.



(b) PSNR in dB

(c) SSIM [Wang et al., 2004]

Figure 4.14: **Denoising results on Berkeley database.** (a) Example images from database. (b),(c) Denoising results for the following models (from left to right): Wiener filter, standard nonlinear diffusion, FoE model, and the two variants from [Portilla, 2006b]. The horizontal axes denote the amount of noise added to the images (PSNR in dB). The error bars correspond to one standard deviation. The yellow asterisks denote cases where the performance differs significantly from that of the FoE model.

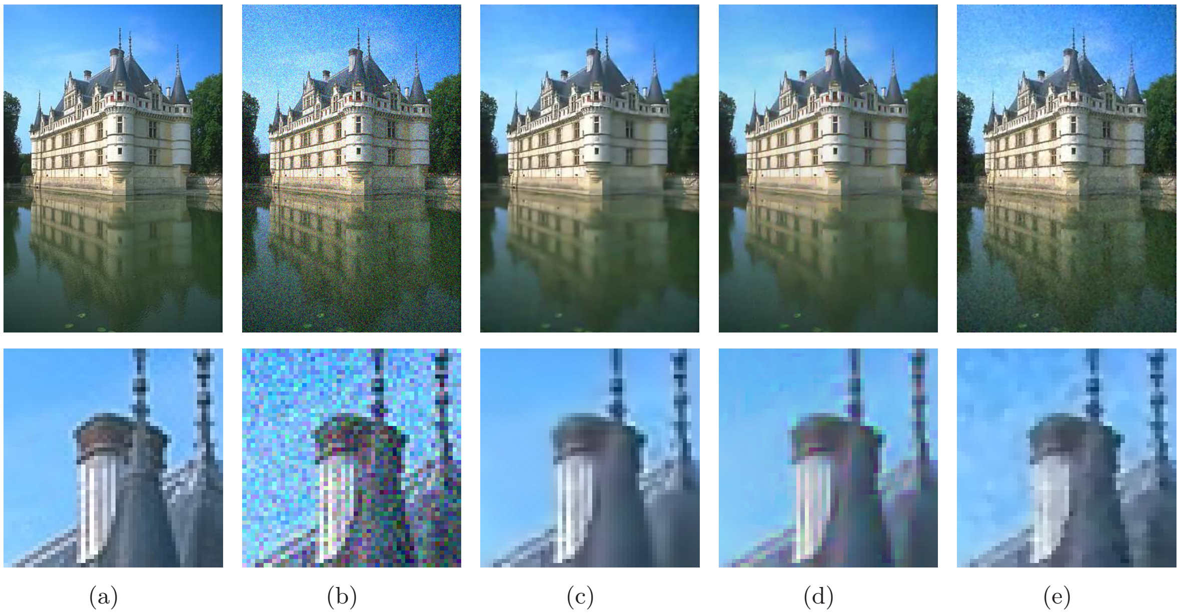

(a) (b) (c) (d) (e)

Figure 4.15: **Color denoising with a Field of Experts.** Full image (top) and detail (bottom). (a) Original noiseless image. (b) Image with additive Gaussian noise added independently in RGB space ($\sigma = 25$); PSNR = 20.50dB. (c) Image denoised using a Field of Experts in YCbCr space; PSNR = 30.13dB. (d) Image denoised using the approach of Portilla et al. [2003] in RGB space; PSNR = 28.37dB. (e) Image denoised using standard nonlinear diffusion in YCbCr space (comparable to pairwise MRF); PSNR = 28.81dB.

4.3.4 Film-grain noise

Before concluding our treatment of image denoising, let us briefly turn to an application of image denoising to image noise as it occurs in real world applications. So far, we have assumed that the noise we are trying to remove is additive, i. i. d. Gaussian noise with a fixed variance. As already mentioned, in real world applications the noise may, for example, not necessarily be Gaussian. One such example is noise from film grain, which occurs in digital scans of photographic film. In other work [Moldovan et al., 2006, 2007] we analyzed the statistical properties of noise from film grain and found that (1) the noise distribution is non-Gaussian because of the limited opacity range that photographic film admits; and (2) that the noise varies spatially depending on the intensity of the underlying, true pixel. In particular, there is more noise in the mid-tones than in very dark or very bright areas (cf. Figure 4.16 *(left top)*). We showed that these properties can be modeled using a learned likelihood model based on an inhomogeneous beta distribution [Moldovan et al., 2006, 2007], where the local noise parameters depend on the intensity of the true underlying pixels. We can combine this learned likelihood model with an FoE prior and perform denoising, or more specifically de-graining, using gradient-based local optimization (see [Moldovan et al., 2007] for more details). This illustrates one of the advantages of the FoE model, because unlike the Wavelet method compared to in the preceding experiments, the FoE can very easily be combined with complex non-Gaussian likelihood models. Figure 4.16 shows an example result of removing film grain using this combination of a learned likelihood model with an FoE prior.

4.3.5 Discussion

As we have seen, Fields of Experts in their use as generic prior model of natural images perform near the current state-of-the-art on the task of image denoising. As far as we are aware, no results of comparable quality have been obtained with other Markov random field based approaches, particularly not with pairwise MRFs. While this is already a very encouraging result, this is not the only aspect that makes FoE models of natural images useful. The very important advantage of FoEs over many of the specialized denoising techniques reviewed above is that they are generic image models, and denoising is only one of the possible applications. The framework behind many of the other denoising algorithms that we mentioned is much more limited and does not directly allow them to be used in other applications (with a few exceptions, such as the work in and related to [Mairal et al., 2006]). Another important advantage is that the FoE can be used with other, non-Gaussian noise models.

4.4 Application: Image Inpainting[5]

In image inpainting [Bertalmío et al., 2000], the goal is to remove certain parts of an image, for example scratches on a photograph or unwanted occluding objects, without disturbing the overall visual appearance. Typically, the user supplies a mask of pixels that are to be filled in by the algorithm. Simple techniques for automatically finding disturbing image artifacts have been developed [Chang et al., 2005], but we do not consider those here. We instead assume that a mask is given by the user and call $\mathcal{M}$ the set of all masked pixels.

To define an appropriate likelihood, we make the following observations: Pixels that are masked as defective have no relation to the true gray value. Assuming that the defective pixels could have any gray value with equal probability, we simply make the likelihood uniform for all masked pixels. Pixels that are not masked should not be modified at all; we can model this using a Dirac delta centered on the pixel value to be preserved. We thus

---

[5] The image data for these experiments was kindly provided by Guillermo Sapiro and Marcelo Bertalmío.

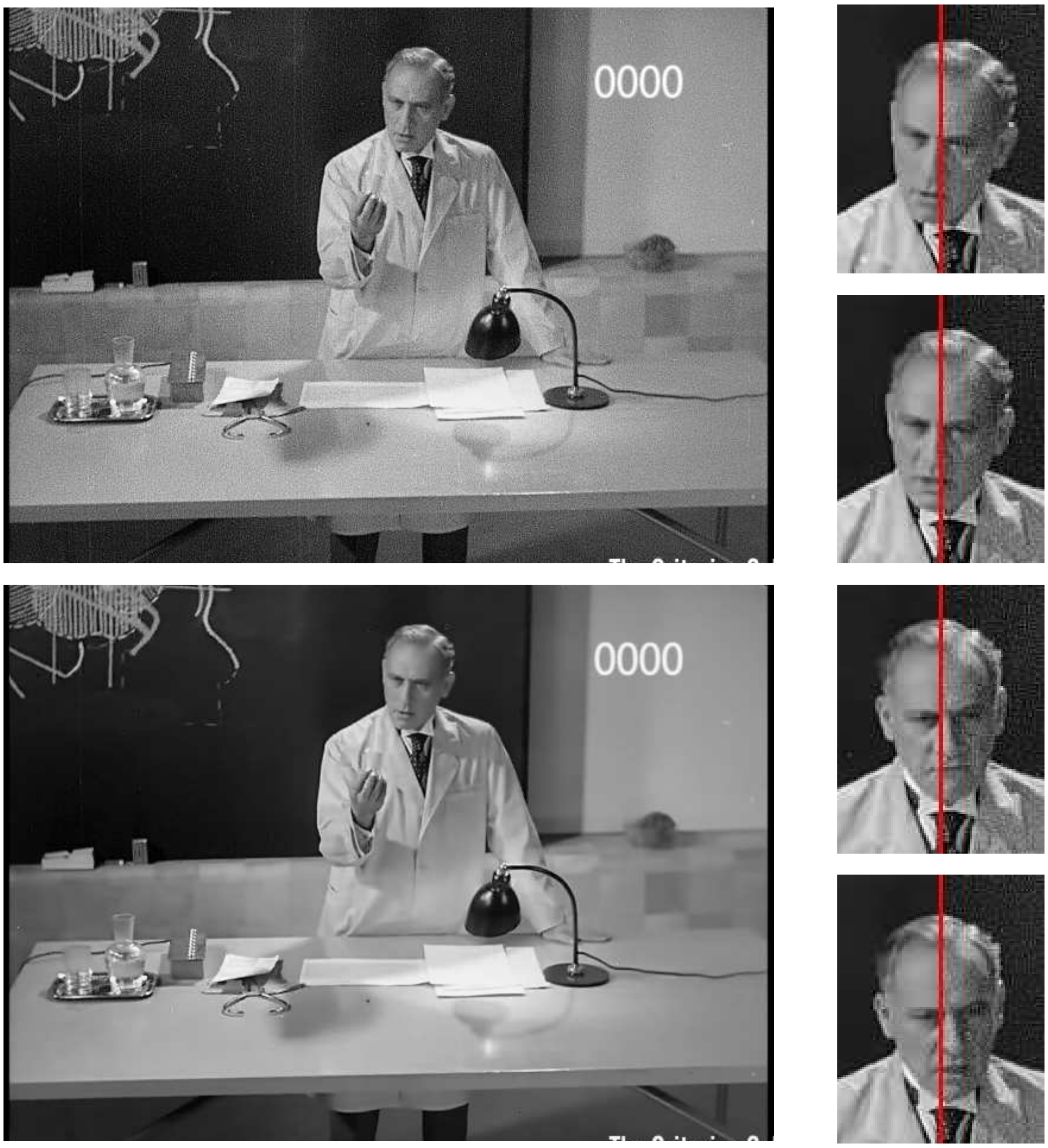


Figure 4.16: **Denoising with an FoE and a learned likelihood model.** *(left top)* Original sequence. *(left bottom)* De-grained sequence. *(right)* Detail results from various frames shown in “split screen” format with the left side being the restored version of the right side.

write the likelihood for image inpainting as

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{M} p(y_i|x_i) \propto \prod_{j=1}^{M} \left\{ \begin{array}{ll} 1, & j \in \mathcal{M} \\ \delta(y_j - x_j), & j \notin \mathcal{M} \end{array} \right\}. \tag{4.13}$$

Past approaches, such as [Bertalmío et al., 2000], use a form of diffusion to fill in the masked pixels. This suggests that the "diffusion technique" we proposed for denoising may also be suitable for this task. To do this, we need to leave the unmasked pixels untouched during gradient ascent, while modifying the masked pixels only based on the FoE prior (since the likelihood is uniform there). We can do this by defining a mask matrix $\mathbf{M}$ that sets the gradient to zero for all pixels outside of the masked region $\mathcal{M}$. Our simple inpainting algorithm propagates information using only the FoE prior:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta \mathbf{M} \left[\sum_{i=1}^{N} \mathbf{J}_i^{-} * \boldsymbol{\psi}'(\mathbf{J}_i * \mathbf{x}^{(t)}; \boldsymbol{\alpha}_i) \right]. \tag{4.14}$$

In contrast to other algorithms, we make no explicit use of the local gradient direction; local structure information only comes from the responses of the learned filter bank. The filter bank as well as the $\boldsymbol{\alpha}_i$ are the same as in the denoising experiments.

Levin et al. [2003] had a similar motivation in that they exploit learned models of image statistics for inpainting. Their approach however relies on a small number of hand-selected features, which are used to train the model on the image to be inpainted. We instead use a generic prior and combine information from many more automatically determined features.

One important limitation of our approach is that it cannot fill in texture, but only "shading". Other techniques have been developed that can also fill in textured areas by synthesizing appropriate textures [Bertalmío et al., 2003; Criminisi et al., 2004].

Figure 4.17 shows the result of applying our inpainting scheme in a text removal application in which the mask corresponds to all the pixels that were occluded by the text. The color image was converted to the YCbCr color model, and the algorithm was independently applied to all 3 channels. Since the prior was trained only on gray scale images, this is obviously suboptimal, but nevertheless gives good results. In order to speed up convergence we ran 5000 iterations of Eq. (4.14) with $\eta = 10$. Since such a large step size may lead to some numerical instabilities, we "cleaned up" the image by applying 250 more iterations with $\eta = 0.01$.

The inpainted result (Fig. 4.17(b)) is very similar to the original and qualitatively superior to those in [Bertalmío et al., 2000]. Quantitatively, our method improved the PSNR by about 1.5dB (29.06dB compared to 27.56dB); the SSIM showed a significant improvement

(a) (b)

(c) (d)

Figure 4.17: **Inpainting with a Field of Experts.** (a) Original image with overlaid text. (b) Inpainting result from diffusion algorithm using the FoE prior. (c) Inpainting result from [Bertalmío et al., 2000]. (d) Close-up comparison between (a) (left), (b) (middle), and (c) (right).

as well (0.9371 compared to 0.9167; where higher is better). Note that to facilitate quantitative comparison with the results of Bertalmío et al. [2000], we measured these results using a GIF version of the input image; [Bertalmío, 2006] confirmed that this was the input for their algorithm. To get a better idea of the performance of the FoE on high-quality input, we also measured results on a JPEG version of the same image. The PSNR was 32.22dB in that case and the SSIM was 0.9736. The advantage of the rich prior can be seen in the continuity of edges which is better preserved compared with [Bertalmío et al., 2000]. Figure 4.17(d) shows a few detail regions comparing our method (center) with [Bertalmío et al., 2000] (right). We can see, for example, that the axle and the wheels of the carriage have been restored very well. Similar qualitative differences can be seen in many parts of the restored image.

Figure 4.18 shows various image inpainting results for test images that were corrupted using synthetic masks (see Section 4.5.12 for more details). An application of this inpainting

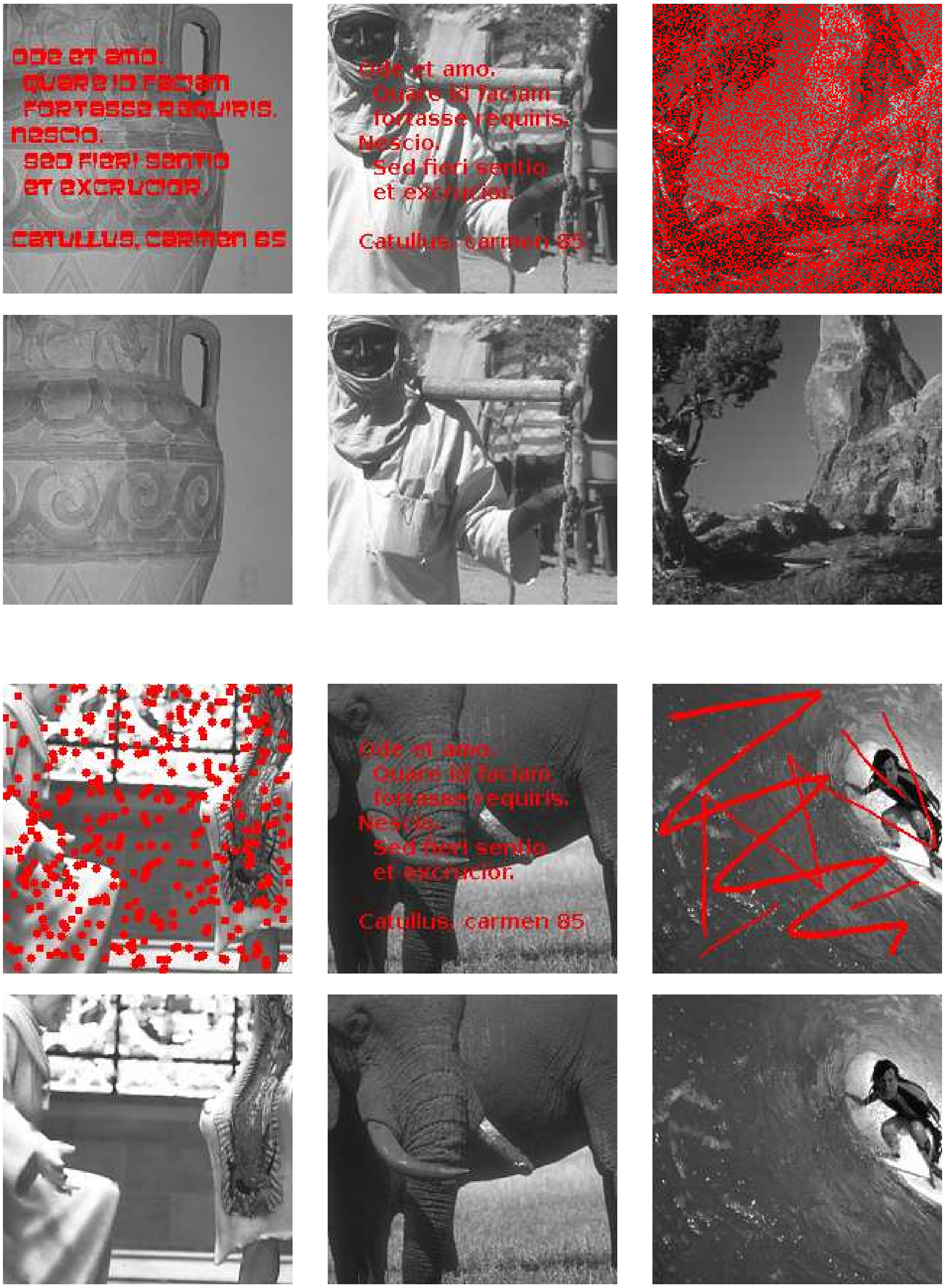


Figure 4.18: **Other image inpainting results.** The first and third rows show the masked images; the red areas are filled in by the algorithm. The second and fourth rows show the corresponding restored images that were obtained using a 5 × 5 FoE model with 24 filters.

algorithm to a problem of scratch removal in a photograph is shown in Figure 1.5(c),(d). Furthermore, Gisy [2005] conducted a detailed study of Fields of Experts in conjunction of image inpainting with generally very encouraging results. The reader is referred to his work for more detailed inpainting experiments.

4.5 Quantitative Evaluation of FoE Parameters

To evaluate the influence of the various parameters and design decisions on the quality of the learned FoE models, we performed a series of experiments. As an example, we varied the size or the number of the filters. As already discussed in Section 3.6, we cannot directly compare the goodness of various models by considering their likelihood, because it is intractable to compute the partition function for FoE models. Instead, we evaluated FoE models in the context of image restoration applications, mostly using image denoising with the same basic setup as in Section 4.3. We also used continuous-space local optimization methods here, hence the performance numbers presented here are not fully indicative of the quality of the FoE model itself, but instead describe the performance of the model in the context of a particular application *and* a particular approximate inference scheme.

The general setup of the experiments was the following: The models were trained on 20000 image patches of 15 × 15 pixels as described in Sections 3.3 and 4.2, except where indicated otherwise. All models suppressed the mean intensity either through choosing an appropriate basis for the filters, or by subtracting the mean from the data and the filters. Except for explicit special cases, the models were initialized with $\alpha_i = 0.01$, and a random set of filters drawn i. i. d. from a unit Gaussian (possibly in a transformed space, as indicated alongside the experiments). If not indicated otherwise, we ran one step of contrastive divergence, where each step is done using hybrid Monte-Carlo sampling with 30 leaps tuned so that the acceptance rate is around 90%. We always performed 5000 iterations of contrastive divergence with a learning rate of 0.01. As a baseline, we used models with 3 × 3 cliques and 8 filters, since those were faster to train and also led to faster inference due to faster convolution operations. In some of the cases, we also considered 5 × 5 cliques with 24 filters.

Once the models were trained, we determined the appropriate λ trade-off parameter for denoising that weighs the FoE prior against the Gaussian image likelihood. We used the same procedure as described in Section 4.3.3.

Using the estimated weight λ^*, every model was evaluated on 68 images from the test portion of the Berkeley segmentation database [Martin et al., 2001] (this is the same set as was used in Section 4.3.3). We added i. i. d. Gaussian noise with $\sigma = 20$ to every image, and subsequently denoised the images with conjugate gradients.

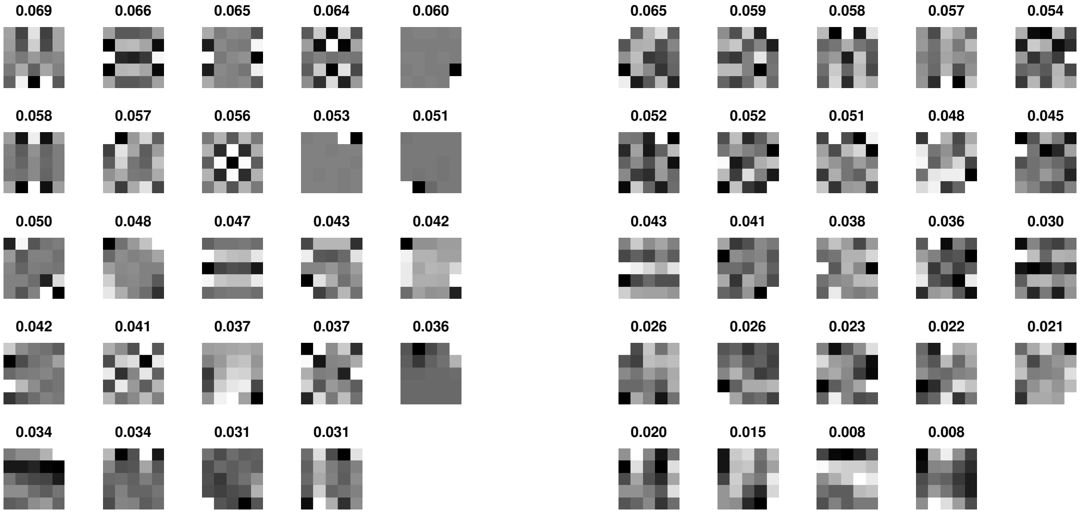
To analyze the FoE model, we evaluated the effects of the following aspects on performance in the respective section:

- 4.5.1: Choice of the filter basis **A**.
- 4.5.2: Size and shape of the filters.
- 4.5.3: Choice of the number of filters.
- 4.5.4: Using either fixed expert parameters, or fixed filter norms.
- 4.5.5: Using fixed, random filters as opposed to learning them.
- 4.5.6: Using fixed filters from patch-based models, instead of learning them.
- 4.5.7: Dependence on the filter initialization.
- 4.5.8: Priors on the filter coefficients.
- 4.5.9: Choice of the expert function.
- 4.5.10: Choice of sampling and learning parameters.
- 4.5.11: Other methods for handling image boundaries.
- 4.5.12: Using linear- and log-intensity images.

In all but Section 4.5.12, we will measure the performance using the described denoising task and give both PSNR and SSIM results averaged over all 68 test images. To reduce the influence of the image boundary on the measurements, we ignore 10 pixels around the boundary when computing the PSNR and SSIM. In Section 4.5.12, we will analyze the application performance on an image inpainting task, where we also used the same 68 test images, but instead of removing noise we filled in pixels according to various masks. More details will be provided alongside the experiment.

4.5.1 Learning the filters in transformed spaces

In this first set of experiments, we evaluated how the choice of the basis **A**, in which the filters $\mathbf{J}_i$ are defined, affects the performance. As we discussed in Section 3.3.4, defining the filters in other bases does not change the actual maximum likelihood objective. But since contrastive divergence learning entails a local gradient ascent procedure, it is susceptible to local optima, and the choice of the filter basis may thus prove important for convergence to a good local optimum. We used three different bases here: (1) A basis that defines the filters in their original space. In this case the basis was just the identity matrix, i. e.,



(a) Model trained in original space ($\mathbf{A}_O$).

(b) Model trained in whitened space ($\mathbf{A}_W$).

Figure 4.19: **Learned model with** 5×5 **cliques and Student t-experts** when trained with different filter bases. The number above each filter denotes the corresponding expert parameter α_i.

$\mathbf{A}_O = \mathbf{I}$. This means that every filter coefficient in this space directly corresponds to a clique pixel. (2) A basis based on whitening the clique pixels. As shown in Section 3.3.4, such a basis is defined as $\mathbf{A}_W = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$ is an eigendecomposition of the covariance matrix $\mathbf{\Sigma}$. This covariance matrix is the covariance of natural image patches that have the same size as the filters. If we choose the matrix of all transformed filters $\tilde{\mathbf{J}}$ as the identity matrix, then the filters $\mathbf{J} = \mathbf{A}_W^{\mathrm{T}}\tilde{\mathbf{J}}$ in the original space are just the principal components scaled according to their standard deviation. This means that the low-frequency principal components have a larger norm than the high-frequency ones, which makes it easier to find low-frequency filters. (3) A basis based on an "inverse" whitening, defined as $\mathbf{A}_I = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$. If we choose the transformed filters $\tilde{\mathbf{J}}$ as the identity matrix in this case, then the filters $\mathbf{J} = \mathbf{A}_I^{\mathrm{T}}\tilde{\mathbf{J}}$ in the original space are also the principal components, but are now scaled according to their inverse standard deviation. In this case the high-frequency principal components have a larger norm than the low-frequency components, which makes it easier to find high-frequency filters. For each of these bases, we evaluated two different cases. In one case we updated the expert parameters α_i directly according to the gradient from Eq. (3.19), in the other case we enforced their positivity by updating their logarithm according to Eq. (3.42). As a reminder, positivity of the α_i is required to make the experts proper distributions.

When training with these three different bases we found filters with high-frequency, and seemingly non-regular structures in all three cases. Figure 4.19(b) shows the filters

Model	3×3, 8 filters					
Filter basis	whitened, $\mathbf{A}_W$		original, $\mathbf{A}_O$		"inverse" whitened, $\mathbf{A}_I$	
Update α	direct	log	direct	log	direct	log
PSNR in dB	27.24	27.34	28.09	28.06	28.40	28.79
SSIM	0.757	0.759	0.784	0.778	0.794	0.813

Model	5×5, 24 filters					
Filter basis	whitened, $\mathbf{A}_W$		original, $\mathbf{A}_O$		"inverse" whitened, $\mathbf{A}_I$	
Update α	direct	log	direct	log	direct	log
PSNR in dB	27.12	25.76	27.90	28.31	28.37	29.07
SSIM	0.773	0.665	0.782	0.792	0.800	0.819

Table 4.2: **Denoising performance of the FoE model when trained with different filter bases.** The filters are trained either in original, whitened, or "inverse" whitened coordinates (see text). In the indicated cases the log of the expert parameters α_i was updated, otherwise the α_i were updated directly.

obtained by learning in a whitened space (corresponding to $\mathbf{A}_W$), which appear the least regular of all and are possibly a result of convergence to a poor local optimum (see below). Figure 4.19(a) shows the results from training in the original space (i. e., with $\mathbf{A}_O$), where the learned filters appear much more regular. Some of the filters look similar to principal components (cf. Fig. 4.5), but others are more localized. Figure 3.4 in the previous chapter shows the filters when training with the "inverse" whitening basis $\mathbf{A}_I$; the learned filters also have high spatial frequencies and are even more localized. While the filters exhibit some qualitative differences depending on the choice of basis, there are even stronger quantitative differences. As Table 4.2 shows, the denoising performance deteriorated when using the whitened basis as opposed to training in the original space. Using "inverse" whitening, on the other hand, led to quantitatively superior results. These findings were consistent for models with 3×3 cliques and 5×5 cliques. Furthermore, we found that updating the logarithm of the expert parameters α_i led to better results in almost all of the cases, sometimes to significantly better results. It is also interesting to note that learning with a whitened basis was relatively unstable when using the standard learning rate of 0.01 (i. e., the filters changed a lot over the course of a few iterations), while learning with the other two bases was quite stable. Most likely, the ascent in whitened space was stuck in a poor local optimum with a wide basin of attraction. This is likely explained by the fact that a whitened basis, which encourages low-frequency components, makes it difficult for high-frequency filters to emerge; but those seem to be important for obtaining good performance.

These quantitative findings strongly suggest that high-frequency filters are important to achieving good performance with FoE models in an image denoising application. As we will see below, experiments with various random filters led to results that are fully consistent

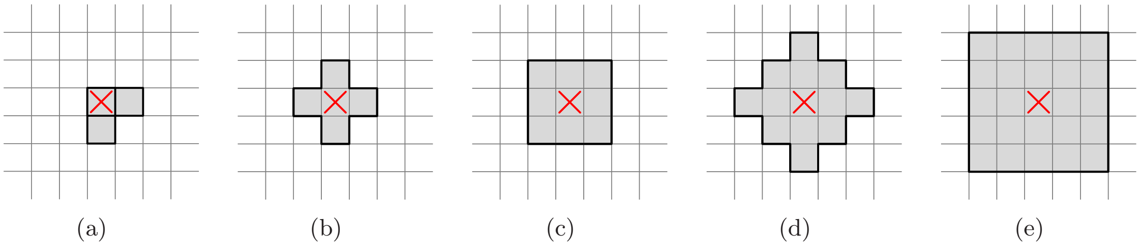


Figure 4.20: **FoE clique structure for various clique shapes and sizes.** (a) 2×1 cliques of pairwise MRF. (b) Diamond-shaped 3×3 clique or fully connected 4-neighborhood. (c) Square 3×3 clique or fully connected 8-neighborhood. (d) Diamond-shaped 5×5 clique. (e) Square 5×5 clique.

with this observation. Since the "inverse" whitening basis encourages high-frequency filters and consistently led to the best quantitative results, we used it as baseline for most of our experiments, unless indicated otherwise. Furthermore, since updating the log of the expert parameters led to better results, we also adopted this as part of the baseline for the remaining experiments.

4.5.2 Varying the clique size and shape

In the second set of experiments, we evaluated how the size and the shape of the maximal cliques influenced the performance of the Fields-of-Experts model. Figure 4.20 shows some of the clique shapes and sizes that were evaluated. The simplest conceivable model based on the FoE framework is the regular pairwise Markov random field shown in Figure 4.20(a), where each node is connected to its top, bottom, left, and right neighbors. Here, there are two types of maximal cliques: pairs of nodes connected by either horizontal or vertical edges. These can be modeled in the FoE framework by restricting 2×2 filters to pairs of horizontal or vertical pixels as depicted in the figure. In Figure 4.20(b), we see a more complicated non-square clique structure, where the 4-neighborhood around a central pixel (marked red) is fully connected. This clique shape was achieved by forcing the filter coefficients of 3×3 filters to be zero outside of the diamond shape. In Figure 4.20(c), we see a simple, square 3×3 clique, where a pixel and its 8 neighbors are all fully connected. We can also also have larger diamond-shaped cliques as shown in Figure 4.20(d), where the filter coefficients of 5×5 filters were forced to be zero outside of the diamond. Finally, in Figure 4.20(e) we can see square 5×5 cliques that were obtained once again by fully connecting all nodes inside the square. Beyond what is shown here, we also evaluated the performance of models with 7×7 filters, both in the diamond-shaped and in the square case. In each case we used the general experimental setup as outlined above, in particular we used "inverse" whitening for defining the filter basis.

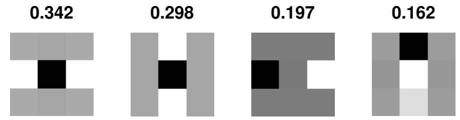
Size	2×1	3×3			
Shape	pairwise	diamond		square	
# of filters	1	4	8	4	8
PSNR in dB	26.58	27.81	27.90	28.63	28.79
SSIM	0.718	0.766	0.769	0.805	0.813

Size	5×5				7×7			
Shape	diamond		square		diamond		square	
# of filters	8	12	8	24	8	24	8	48
PSNR in dB	28.81	28.88	28.80	29.07	28.78	28.98	28.74	29.04
SSIM	0.815	0.816	0.811	0.819	0.811	0.817	0.812	0.818

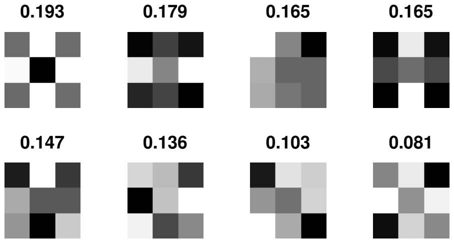
Table 4.3: **Denoising performance of FoE models with various clique sizes and shapes.**

In each of the cases, we evaluated the performance using two experiments: First, we trained and tested models with a fixed number of 8 filters. Then we trained and tested models with $p - 1$ filters, where p is the number of nodes in the maximal cliques. Since, as for all the experiments in this chapter, we ignored the mean gray value component of each clique-sized patch, this means that there were as many experts per clique as there are degrees of freedom. Figures 4.21 and 4.22 show some of the learned models (see also Figure 3.4 for a 5×5 model with square cliques). Table 4.3 gives the performance measurements from these experiments.

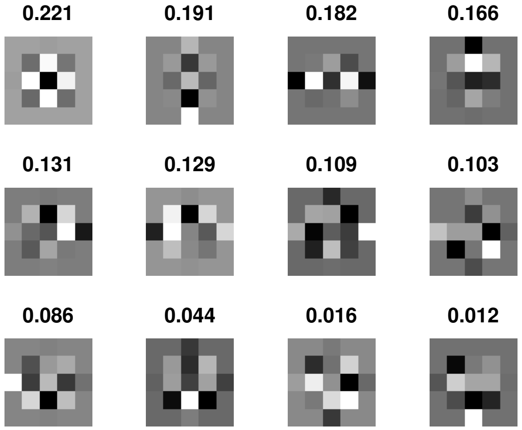
We can see that the pairwise model performed substantially worse than the high-order models with square cliques. This again showed that FoEs with large cliques are able to capture rich structures in natural images that cannot be captured using pairwise MRF models alone. In the 3×3 case, square cliques substantially outperformed diamond-shaped ones. For 5×5 and 7×7 FoEs, diamond-shaped cliques performed on par with square ones with few filters, but performed worse than square ones with many filters. Models with square 3×3 cliques and 8 filters already performed quite well, but a 5×5 model with 24 filters nevertheless outperformed the simpler model by a considerable margin. A 7×7 FoE with 48 filters was able to match the performance of the 5×5 model with 24 filters, but did not exceed it. Interestingly, the performance of models with larger cliques (5×5 and 7×7) was best when many filters are used; with only 8 filters the 3×3 FoE was superior. We conclude that while cliques larger than 3×3 improved performance and captured more structure of natural images, models with square 3×3 cliques already captured a large amount of the variation in natural images, at least of the variation that can be captured with linear projections and Student t-experts. It is conceivable that this could be improved on with other experts, or with experts that model nonlinear features of the clique pixels,



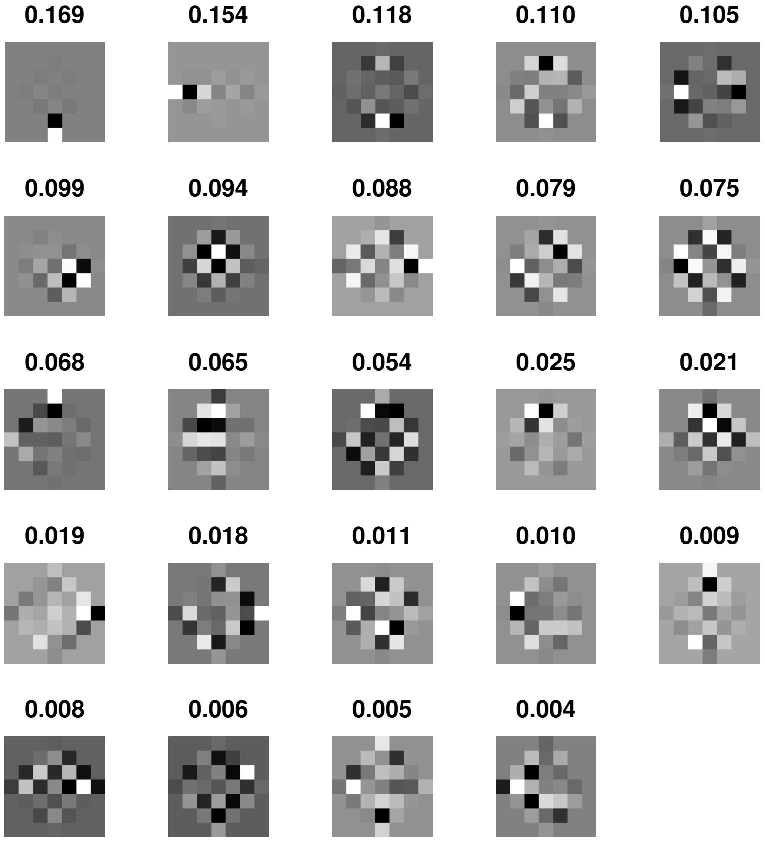
(a) Diamond-shaped 3×3 cliques with 4 filters.



(b) Square 3×3 cliques with 8 filters.



(c) Diamond-shaped 5×5 cliques with 12 filters.



(d) Diamond-shaped 7×7 cliques with 24 filters.

Figure 4.21: **Learned models with various clique sizes and shapes.** The number above each filter denotes the corresponding expert parameter α_i.

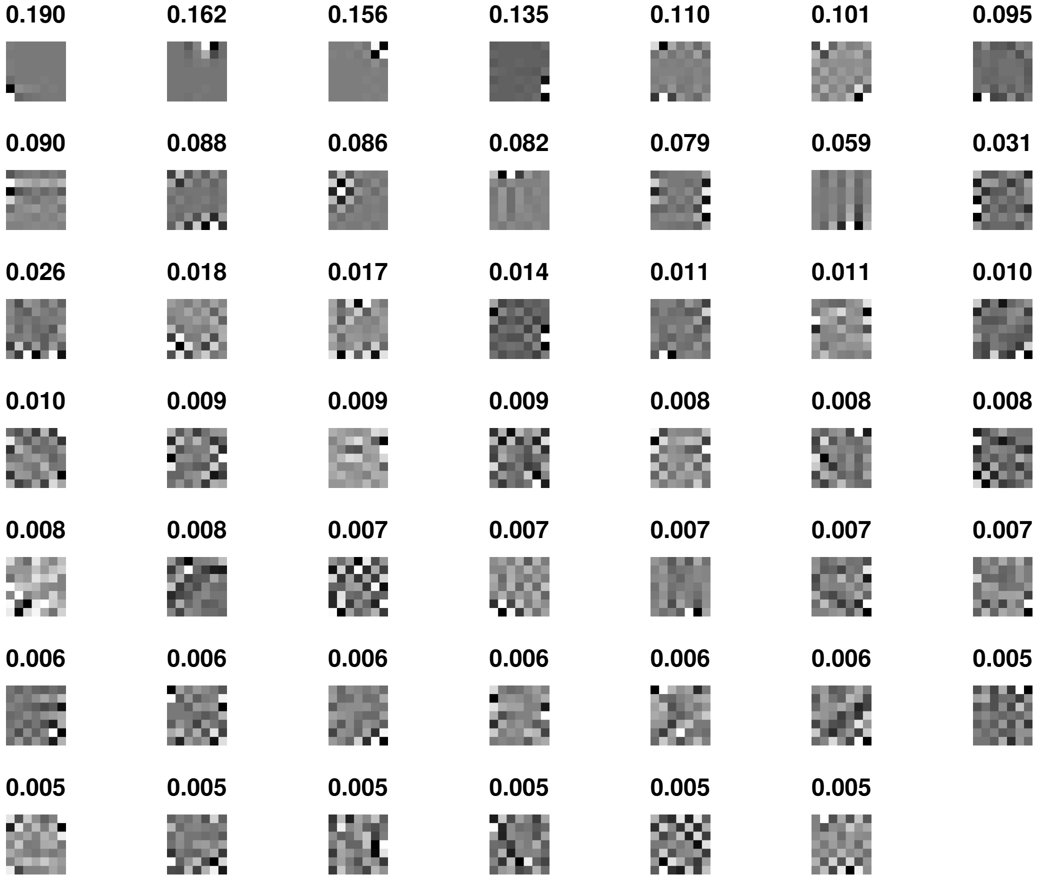


Figure 4.22: **Learned model with 7×7 square cliques and 48 filters.** The number above each filter denotes the corresponding expert parameter α_i.

but explorations of this are left for future work.

4.5.3 Varying the number of filters

The next set of experiments determined the impact of the number of filters on the denoising performance. To that end we trained and tested 3×3 models with 1 through 18 filters otherwise using a baseline setup. Figure 4.23(a) graphs the denoising performance as a function of the number of experts, measured both in terms of PSNR and SSIM and, as usual, averaged over all 68 images from the test set. We can see that about four to five filters were necessary to make the model perform well, and that the performance improvements became quite minor beyond eight filters. Nonetheless, there were small improvements in test set performance even for large numbers of filters, which is an important indication that we were not overfitting the training data.

In Figure 4.23(b) we show the sum of the expert parameters (i. e., $\sum_i \alpha_i$) plotted for each number of filters. This illustrates that even though more filters may be used, their

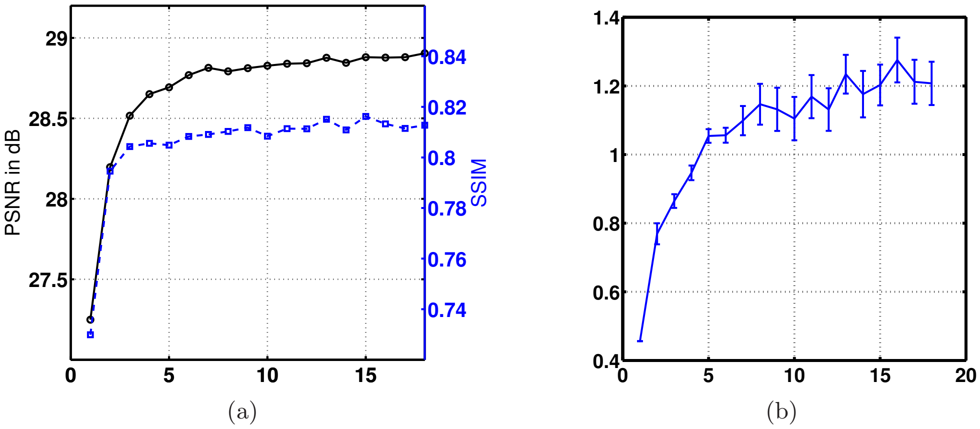


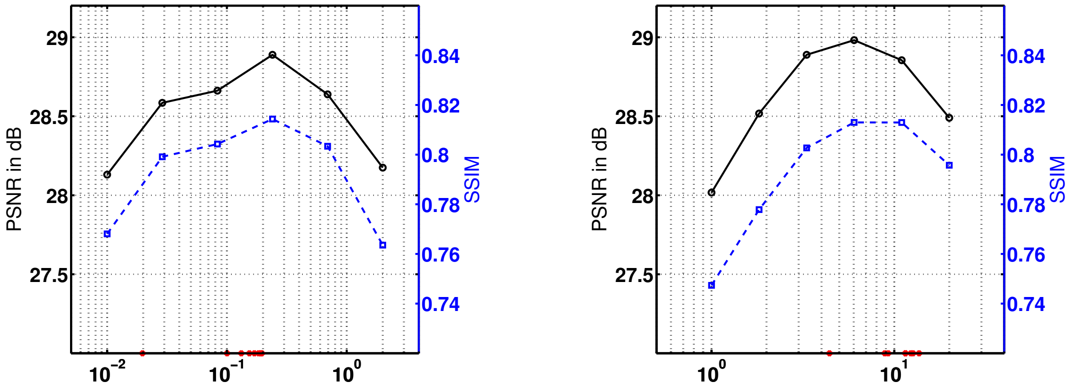
Figure 4.23: **Denoising performance of 3×3 models with a varying number of filters.** Part (a) shows the performance for a given number of filters in terms of PSNR (solid black, circular markers) and SSIM (dashed blue, square markers). Part (b) shows the sum of the α_i for each number of filters, which illustrates how the "weight" gets distributed to more filters. The error bars indicate the standard deviation of the α_i.

combined "weight" did not change significantly, but instead was distributed across more filters.

4.5.4 Using fixed expert parameters or fixed filter norms

In order to determine how robustly the learning algorithm finds suitable parameters, we next studied the performance of models where some of the model parameters are fixed instead of learned. In the first part of the experiment we set all expert parameters α_i to a constant and only learned the filters. This could easily be done by simply not updating the expert parameters during training. We tried six different values for α_i in the range $[0.01, 2]$ and spaced them evenly in the log-domain. In the second part we fixed the norm of all filters $\mathbf{J}_i$ to some value, and learned the direction of the filter vector (i. e., the pixel pattern that it is describing) as well as the expert parameters. We used six values for the filter norms taken from the range $[1, 20]$ and evenly spaced them in the log-domain. In this case we performed training by updating the filters with the gradient expression from Eq. (3.21) (in "inverse" whitened space), and after each iteration setting the norm of the filters back to the intended value. This effectively projects the filter gradient onto the manifold of filter updates that do not affect the norm of the filters.

Figure 4.24(a) shows the results for a 3×3 model with 8 filters and fixed expert parameters. We can see that the performance was rather sensitive regarding the setting of the expert parameters, since the performance fell off quite steeply from its peak. Furthermore, the peak value exhibited a slightly worse performance than when learning the expert parameters, which suggests that fixing the expert parameters is not a sensible way of regularizing



(a) Fixed expert parameters.

(b) Fixed filter norm.

Figure 4.24: **Denoising performance of 3×3 models with 8 filters and fixed filter norm or fixed expert parameters.** All filters have either the same filter norm, or the same expert parameter as indicated by the value on the x-axis. The red stars on each x-axis indicate the expert parameters or filter norms that were determined automatically in the fully learned case.

the learning procedure. The red stars on the x-axis of the graph show the expert parameters that were determined automatically when the training procedure was not restricted by setting all of them to the same, fixed value.

In Figure 4.24(b) we see the denoising performance for fixing the filter norms in otherwise the same model (note that the expert parameters were learned in this case). The red stars on the x-axis of the graph indicate the filter norms that were determined automatically in case of a fully learned model. We can see that the performance was also somewhat sensitive to the norm of the filters, although the sensitivity appeared not to be as pronounced as that regarding the expert parameters. Furthermore, the performance at the peak value slightly exceeded the performance of the normal training procedure where all parameters were learned from data. This suggests that fixing the norm of the filters helps regularizing the learning algorithm to some extent. Nonetheless, fixing the filter norm requires determining the optimal operation point using a large set of experiments and is thus quite tedious in practice. Because of that, we did not adopt this procedure in any of the other experiments here.

4.5.5 Using fixed, random filters

In order to determine the effect of learning the filters, we performed two kinds of experiments. Here, we compared against fixed, random filters and only learned the expert parameters and the norm of the filters (while keeping their direction fixed). In the following section, we will compare against using fixed filters that have been determined ahead of time

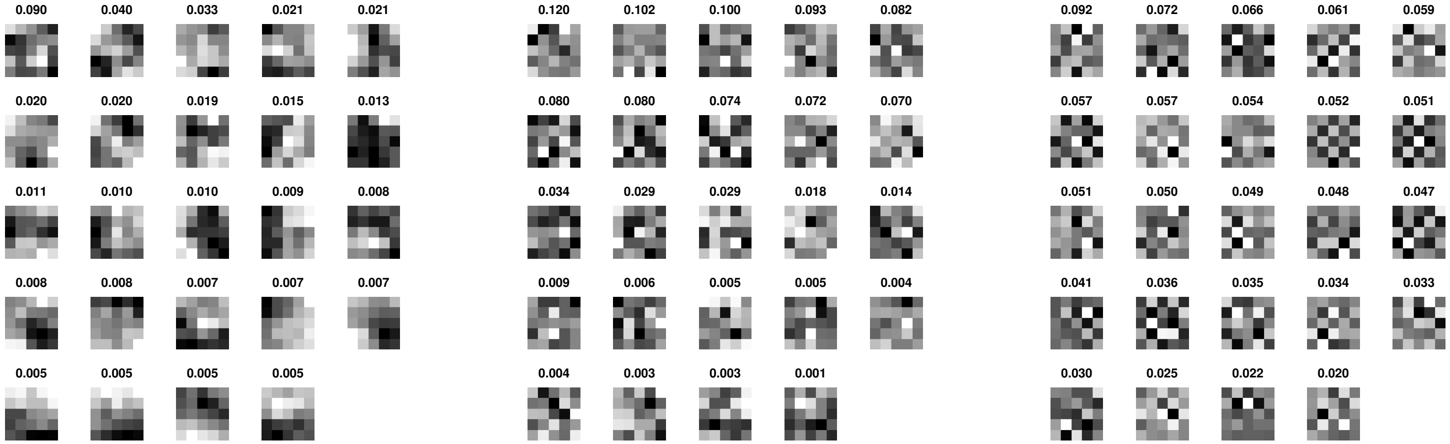
Model	3×3, 8 filters					
Space	whitened		original		"inverse" whitened	
Initialization	unnormalized	normalized	unnormalized	normalized	unnormalized	normalized
PSNR in dB	26.61	26.48	27.28	27.30	27.80	27.80
SSIM	0.746	0.736	0.761	0.762	0.779	0.779

Model	5×5, 24 filters					
Space	whitened		original		"inverse" whitened	
Initialization	unnormalized	normalized	unnormalized	normalized	unnormalized	normalized
PSNR in dB	25.70	25.96	26.57	26.70	27.99	27.99
SSIM	0.694	0.714	0.748	0.749	0.783	0.783

Table 4.4: **Denoising performance of the FoE model with random filters.** The filters are drawn either in original, whitened, or "inverse" whitened coordinates (see text). In the indicated cases the random filters were normalized to unit norm before the actual training procedure.

using another method. For the random filter experiments we worked with three basic setups corresponding to different filter bases: (1) We first drew filters randomly in the original space (corresponding to $\mathbf{A}_O$ from above) by drawing all coefficients $J_{i,j}$ of the matrix $\mathbf{J}$ of all filters i. i. d. from a unit normal (i. e., $J_{i,j} \sim \mathcal{N}(0,1)$). (2) In the second case we drew filters randomly in whitened space so that $\mathbf{J} = \mathbf{A}_W^{\mathrm{T}} \tilde{\mathbf{J}}$, where $\tilde{J}_{i,j} \sim \mathcal{N}(0,1)$. Typical random filters obtained in this way look quite smooth, because the low-frequency principal components are scaled up by their (large) standard deviation. (3) In the final case we drew filters randomly using "inverse" whitening. Here, $\mathbf{J} = \mathbf{A}_I^{\mathrm{T}} \tilde{\mathbf{J}}$, where the coefficients of the transformed filter matrix $\tilde{\mathbf{J}}$ were again drawn from a unit normal. In this case, the typical filter samples were dominated by high-frequency structures, because the high-frequency principal components are scaled up in this case due to their small standard deviation. In all three cases we tested both 3×3 models with 8 filters and 5×5 models with 24 filters. Beyond that, we tried two different variants each, one in which we started the training procedure from the filters as sampled, and one where we normalized all random filters to unit norm prior to starting the training procedure. As a reminder, training here means learning the expert parameters and the norms of the filters. The norms of the filters were trained by first updating the filters using the gradient from Eq. (3.20) and then applying the change of the filter norm from this update to the fixed filter that we intend to use. This projects the filter gradient onto the manifold of matrices with fixed direction column vectors (i. e., filters), but variable filter length. Note that since we did not learn the filter direction in this experiment, we did not use any form of filter basis *during* training. Figure 4.25 shows learned 5×5 models for each of the 3 different filter bases.

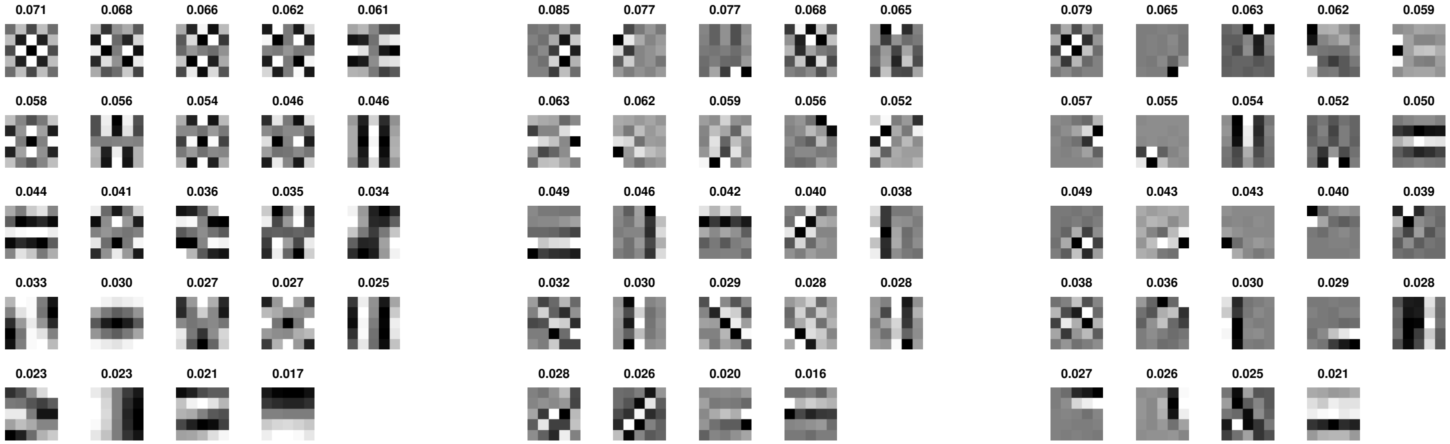
The quantitative results shown in Table 4.4 once again underline the importance of



(a) Model with random filters in whitened space.

(b) Model with random filters in original space.

(c) Model with random filters in "inverse whitened" space.



(d) Model with PCA filters.

(e) Model with ICA filters.

(f) Model with PoT filters.

Figure 4.25: **Learned FoE models based on various kinds of fixed filters.** Each model has 5 × 5 cliques and 24 filters. The number above each filter denotes the corresponding expert parameter α_i. The filters are sorted according to their expert parameters in descending order.

high-frequency filters for achieving good denoising performance with Fields of Experts. Filters drawn in whitened space performed poorly, particularly in case of the 5×5 model. Filters drawn in the original space performed better, but still not nearly as well as filters drawn in the "inverse" whitened space, which emphasizes high-frequencies with a certain structure. It is also interesting to note that this space was the only one where the 5×5 model outperformed the corresponding 3×3 model with random filters. When we compare the results to those in Table 4.3, we see that even with "inverse" whitening the performance with random filters was about 1dB short of that with learned filters. While the model generally performed well with random filters (at least with "inverse" whitening), learning the filters substantially improved performance. We also note that normalizing the filters prior to learning did not have noticeable effects in the interesting cases.

4.5.6 Using fixed filters from patch-based models

The next set of experiments was similar to the previous one in that we used a fixed set of filters, and only trained the expert parameters as well as the norm of the filters. Instead of initializing with randomly drawn filters, we initialized them with the filters obtained by various patch-based learning methods: (1) First, we used filters obtained by principal component analysis of image patches with the same size as the cliques. (2) We then used filters obtained by independent component analysis of image patches. In particular, we used the software by Gävert et al. [2005] using the standard settings and extracted 8 independent components for 3×3 patches and 24 for 5×5 patches. These filters obtained by ICA are Gabor-like oriented derivative filters at various scales, orientations, and locations. These kinds of filters are commonly used in image processing and are also used in the FRAME model [Zhu et al., 1998] for modeling textures. (3) Finally, we used filters obtained by training a Product-of-Experts model with Student t-experts as described by [Teh et al., 2003]. The filters obtained by this PoT model also mostly look like oriented derivative filters (cf. Fig. 2.6). Starting with these fixed filters, learning proceeded as in the previous section by adapting only the norm of the filters and the expert parameters. Here as well, we used both the filters as determined by the patch-based model and normalized versions for initializing contrastive divergence. Figure 4.25 shows the learned 5×5 models with 24 filters for all three cases.

Table 4.5 shows the results from using these fixed filters. We found that PCA filters worked relatively poorly, which was not a surprise given that random filters drawn in the same space also did not perform very well. Nonetheless, the performance was better than with random filters drawn in the same space, which may be attributable to the fact that principal components contain high-frequency filters. Filters from independent component

Model	3 × 3, 8 filters					
Filters	PCA		ICA		PoT	
Initialization	unnormalized	normalized	unnormalized	normalized	unnormalized	normalized
PSNR in dB	27.86	28.02	28.02	28.02	28.12	28.12
SSIM	0.782	0.781	0.781	0.781	0.783	0.784

Model	5 × 5, 24 filters					
Filters	PCA		ICA		PoT	
Initialization	unnormalized	normalized	unnormalized	normalized	unnormalized	normalized
PSNR in dB	28.08	28.31	28.37	28.36	28.51	28.51
SSIM	0.782	0.784	0.790	0.790	0.791	0.790

Table 4.5: **Denoising performance of the FoE model with filters determined by PCA, ICA, and PoE.** In the indicated cases the fixed filters were normalized to unit norm before the actual training procedure.

analysis worked slightly better, particularly in the 5 × 5 case. Finally, filters obtained from a PoT model worked best in this comparison, but still performed between 0.4 and 0.6dB worse that fully learned filters. This really underlines the importance of learning the filters in conjunction with a high-order MRF model such as the FoE. It is furthermore very revealing to closely examine the learned models shown in Figure 4.25. As we can see, high-frequency filters had the highest weight (expert parameter) in all three cases, and regular looking filters consistently got smaller weights. In particular, it is interesting how in case of the PCA filters the sorting based on decreasing weight is almost exactly the reverse of the singular value-based sorting in Figure 4.5. This says that minor components were assigned more weight, and were thus more important than major components. These findings suggest that smooth derivative (e. g., Gabor) filters are not the best choice in conjunction with such a framework and that the kinds of filters found by the full learning algorithm are important for getting good performance. It is interesting to note here that Zhu et al. only relied on Gabor filters in the context of modeling particular classes of textures [Zhu et al., 1998]. In the context of modeling generic image priors, Zhu and Mumford [1997] relied on derivative filters based on neighbor differences (albeit at multiple scales).

4.5.7 Repeated experiments with different initializations

In order to determine the influence of a particular filter initialization on the quality of the model and its effect on denoising results, we evaluated various initializations for a 3 × 3 FoE with 8 filters. This may be important, since the contrastive divergence algorithm is based on a local gradient ascent, and depending on the initialization may end up in different local optima. Similar to using fixed random filters as above, we drew random

Model	3×3, 8 filters					
Filters	random, whitened space			random, original space		
Trial	1	2	3	1	2	3
PSNR in dB	28.01	28.16	28.08	28.44	28.35	28.48
SSIM	0.786	0.787	0.785	0.794	0.790	0.794

Model	3×3, 8 filters					
Filters	random, "inverse" whitened			PCA	ICA	PoT
Trial	1	2	3	1	1	1
PSNR in dB	28.73	28.65	28.75	28.69	28.79	28.71
SSIM	0.803	0.803	0.805	0.805	0.811	0.802

Table 4.6: **Denoising performance of the FoE model with different initializations.**

filters for initialization in three different spaces: in the original space, in whitened space, and in "inverse" whitened space. We used three different sets of random filters in each case. Furthermore, we also initialized with the patch-based filters as in the previous section, i. e., using principal components, independent components, and PoT filters. As opposed to the above experiments, we only used these filters (including the random ones) for initialization, but learned the filters starting from there.

The denoising results in Table 4.6 show that initializing with random filters in the whitened space led to the worst results, while initializing with random filters in the "inverse" whitened space led to better results than when using the original space. In each case the results across the 3 different trials were quite consistent. This suggests that the filter learning procedure does indeed depend somewhat on the initialization, but as long as the filters have the seemingly important characteristics from the high-frequency initialization (in the "inverse" whitened space), the particular initial set of filters is not that important. This encourages us to adopt random filters in the "inverse" whitened space as the baseline; they were used in all experiments unless noted otherwise. We also see from Table 4.6 that initialization with PCA, ICA, or PoT filters did not lead to substantially differing results, both among themselves and also compared to the baseline random initialization.

4.5.8 Priors on filter coefficients

In the next experiment, we determined if there is a performance advantage from regularizing the learning procedure by putting a prior on the filter coefficients. This could also have the effect of simplifying the learned filters. In particular, we put a Gaussian prior on the coefficients (in the transformed space):

$$p(\tilde{J}_{i,j}) = \mathcal{N}(\tilde{J}_{i,j};\ 0, \sigma_J). \tag{4.15}$$

Model	3 × 3, 8 filters			5 × 5, 24 filters		
σ_J	1	3	5	1	3	5
PSNR in dB	28.54	28.86	28.91	28.52	28.96	29.07
SSIM	0.783	0.805	0.811	0.782	0.808	0.817

Table 4.7: **Denoising performance of the FoE model with Gaussian coefficient prior** with given standard deviation σ_J.

This prior can be quite easily integrated into the learning procedure by computing the gradient of the log-prior

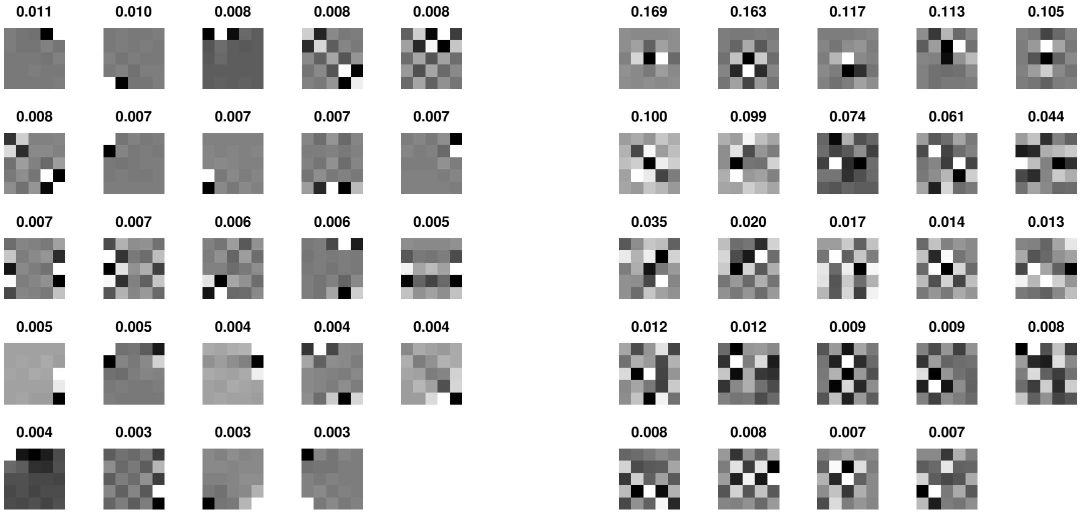
$$\nabla_{\tilde{\mathbf{J}}} \log p(\tilde{\mathbf{J}}) = -\frac{1}{\sigma_J^2}\tilde{\mathbf{J}} \tag{4.16}$$

and adding it to the approximate gradient of the log-likelihood (cf. Eqs. (3.21) and (3.39)). We tried three different values for the standard deviation (1, 3, and 5) in 3×3 and 5×5 models. Alternatively, it would also be possible to put a heavy-tailed prior on the coefficients. This would have the effect of making more of the filters close to zero.

Table 4.7 shows the denoising results from training FoEs with this coefficient prior. We can see that strong priors ($\sigma_J = 1$) led to a deterioration of the results. The weaker priors led to some small performance improvement for the 3 × 3 model, but not with the 5 × 5 models. Overall, there may be a small benefit to putting a weak prior on the filter coefficients in order to regularize the learning procedure, but since the effect was relatively minor, we decided not to use this prior as part of the baseline algorithm.

4.5.9 Charbonnier expert

One interesting question is whether expert functions other than the Student t-expert are useful in conjunction with modeling natural images. To that end we also analyzed the Charbonnier expert as introduced in Section 3.2.1, which is essentially a differentiable version of an exponential distribution (or an L1-norm, when viewed as energy). One advantage of the Charbonnier expert is that its energy is convex, which makes optimization in the context of denoising much easier. The Gaussian likelihood model we are using here is convex as well (more precisely the associated energy), which makes the posterior energy convex. This means that we can actually find global optima during denoising using the conjugate gradient algorithm used here. We trained both 3 × 3 models with 8 filters as well as 5 × 5 FoE with 24 filters. Each kind of model was trained with the filters defined in the original space as well as the filters defined in the "inverse" whitened space. Except for the different expert function training was largely identical to the Student-t case. Figure 4.26(a) shows the filters obtained by training a 5 × 5 model in the original filter space. Similar to the Student-t case, we found high-frequency filters that were somewhat irregular. Nevertheless,



(a) 5 × 5 FoE model with Charbonnier experts.

(b) 5 × 5 FoE model with Student t-experts trained by conditioning on the boundary pixels.

Figure 4.26: **Filters learned for two different** 5×5 **FoE models.** The number above each filter denotes the corresponding expert parameter α_i.

they seem to be more localized that what we obtained using Student t-experts; many of the filters resemble simple derivative filters.

Table 4.8 shows the denoising results obtained with these Charbonnier experts. We can see that the model performed best when trained in the original space, and that the performance with 3×3 and 5×5 cliques was virtually identical. In either case, the performance was worse than with Student t-experts, in case of 5×5 filters quite substantially. This result reinforces the observation that non-convex regularization methods are necessary for achieving good performance in low-level vision applications [e. g., Black et al., 1998]. While this obviously makes optimization more difficult, performance already improved from using non-convex models even when relying on simple local optimization methods. It is interesting to note, however, that denoising using the Charbonnier expert proceeded much more quickly, as many fewer conjugate gradient iterations were needed in practice. In case of Student t-experts, denoising often required 3000 or more iterations, but with Charbonnier experts denoising often required as few as 250 to 350 iterations until convergence. In applications, where computational performance is crucial, it may thus be interesting to use FoE models with convex experts (when they are viewed as energies).

4.5.10 Sensitivity to sampling and learning parameters

Another interesting aspect to investigate is whether the results depend on the type of sampler, or the parameters of the sampling and learning procedure. We focused on two

Model	3×3, 8 filters		5×5, 24 filters	
“Inverse” whitening	N	Y	N	Y
PSNR in dB	28.69	28.52	28.74	28.47
SSIM	0.806	0.792	0.808	0.790

Table 4.8: **Denoising performance of the FoE model with Charbonnier experts.**

Model	3×3, 8 filters			5×5, 24 filters
l	5	25	1	5
Sampler	HMC		Gibbs	HMC
PSNR in dB	28.67	28.60	28.79	28.84
SSIM	0.805	0.802	0.809	0.808

Table 4.9: **Denoising performance of the FoE model with different sampling and learning parameters.**

aspects here: First, we evaluated whether running contrastive divergence for more than $l = 1$ steps improves performance. Running CD for more steps makes the objective more similar to maximum likelihood, which allows us to see how much performance is traded off for having a faster learning procedure. Second, we evaluated if using a Gibbs sampler leads to different results than using the hybrid Monte Carlo approach adopted otherwise. For the first experiment, we trained a 3×3 model with 8 filters using $l = 5$ and $l = 25$ steps per contrastive divergence iteration (i. e., iteration of the learning algorithm). Furthermore, we trained a 5×5 model with 24 filters using $l = 5$ contrastive divergence steps. For the second experiment, we used a simple Gibbs sampler that updates one pixel at a time using the conditional distribution computed for all values in $\{0, \dots, 255\}$, and trained a 3×3 model for efficiency reasons (here we used $l = 1$ CD step per iteration).

We should note that the runtime scales proportionally with the number of steps. This means that using $l = 25$ steps became almost impractically slow (it took several weeks to train the model). This further illustrates the importance of using contrastive divergence as principled replacement for maximum likelihood estimation.

Table 4.9 shows the denoising results for the various models. We can see that running CD for more than 1 step actually led to models with a slightly deteriorated denoising performance, although the difference was not drastic. The Gibbs sampler, even though much slower than hybrid Monte Carlo sampling, did not lead to any considerable performance difference to training with HMC. Overall, the exact choice of sampling algorithm or of the number of contrastive divergence steps did not seem particularly critical to obtaining good performance, but it is nevertheless interesting to note that $l = 1$ contrastive divergence steps led to models with the best denoising performance.

Model	3×3, 8 filters		5×5, 24 filters	
Image size	15×15	17×17	15×15	19×19
PSNR in dB	28.73	28.78	29.12	29.01
SSIM	0.808	0.808	0.819	0.815

Table 4.10: **Denoising performance of the FoE model conditioned on boundary pixels** (see text).

4.5.11 Alternative boundary handling[6]

Next we evaluated if it is possible to improve the performance of the FoE model in denoising tasks by using alternative ways of handling the image boundaries (during training). As discussed in Section 3.6, we typically define the FoE model so that it only contains cliques that fully overlap with the support of the training images. As we noted, this seems to create a bias toward large filter coefficients at the boundary of the filter mask, because the pixels at the image boundary are constrained by fewer cliques. Earlier, we also proposed and discussed one possible solution, in particular to train the model conditioned on the boundary pixels (cf. Eq. (3.59)). We carried out this conditional training procedure by sampling only the pixels that were sufficiently far away from the boundary of the training images, and otherwise used a baseline learning procedure. We trained 3×3 and 5×5 models with our usual training image size of 15×15 pixels, as well as slightly larger training images where during training we sampled 15×15 pixels in the interior.

Figure 4.26(b) shows the filters that were learned using this conditional training procedure. As we can see, the filters appear somewhat more regular despite still having high spatial frequencies. In particular some of the filters look similar to localized *second* derivative filters (the first and last filter in the first row). Nevertheless, many of the filters, even those with large weights, still look very much unlike Gabor-like filters found using patch-based models. Table 4.10 gives the quantitative results from denoising with this model. Note that during denoising we ignored the conditional nature of the model and treated the model as if it was trained in a fully generative fashion. Nonetheless, as usual we ignored 10 pixels around the boundary for measuring the performance, so the effects of this slightly inconsistent use are likely to be very minor. The denoising performance seemed to be virtually identical to that of models trained in the usual, fully generative way.

---

[6] This experiment was kindly suggested by Pietro Berkes.

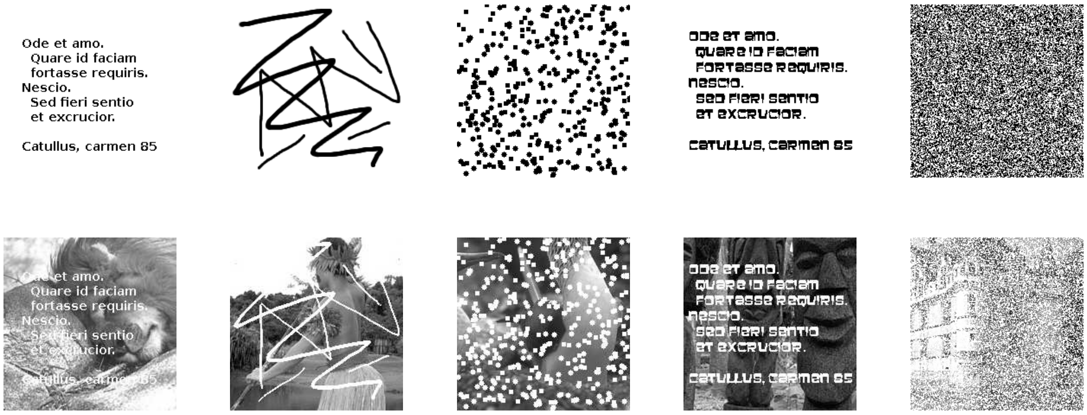


Figure 4.27: **Inpainting masks used to generate benchmark set (top row).** Black indicates corrupted image regions. The bottom row shows corrupted example images used for testing.

4.5.12 Other image data

Finally, we evaluated whether it is advantageous to model linear-intensity images or log-intensity images instead of the gamma-compressed images that we used for all other experiments in this chapter. To that end we transformed the gamma-compressed original data into linear- and log-intensities as discussed in Section 4.1.2. We then trained 3×3 and 5×5 FoE models in the usual fashion. The only change (apart from the data) was that the patch-covariance that builds the basis for the "inverse" whitening transformation was determined separately for each type of data. The filters learned for linear- and log-data are qualitatively quite similar to those learned from gamma-compressed data, and are thus not shown here.

We nevertheless would like to measure performance and compare the three different models. For that we chose an inpainting task, because the inpainting likelihood from Eq. (4.13) is invariant to any transformations of the intensity domain. This makes it very easy to apply, as inpainting can be carried out in the native domain of the respective model. We created an inpainting benchmark by taking a 200×200 region from each of the 68 images in our test set and corrupting pixels inside a clutter mask. There are five different masks including text, hand-drawn scribbles, and salt-and-pepper noise (see Figure 4.27). Each of the 68 test images was corrupted based on one of the five masks, so that each mask was roughly used equally often; the pixels to be corrupted were simply set to a constant intensity (see Figure 4.27 for example images). Inpainting was done just as described in Section 4.4.

While inpainting itself is easy even if we work in a different data domain, it is actually not that straightforward to compare the results, since to have a useful comparison they have to be compared in the same domain. We could argue that the gamma-compressed domain in which digital images are typically stored is perceptually the most relevant domain, and if we

Model	3 × 3, 8 filters			5 × 5, 24 filters		
Image data	gamma	linear	log	gamma	linear	log
PSNR in dB	31.81	27.69	32.02	32.59	27.87	32.52
SSIM	0.944	0.900	0.945	0.949	0.902	0.948

Table 4.11: **Inpainting performance of the FoE model trained on various types of image data.** The gamma-compressed data was transformed into linear or log-intensities, if necessary, before inpainting. After inpainting the results were transformed back to the gamma-compressed domain, where the image quality was evaluated.

ultimately care about perceptually good results, then we should compare the results there. Nonetheless, this may introduce an unfair bias favoring the model of gamma-compressed data, but there does not seem to be an elegant workaround. Hence, we computed both PSNR and SSIM after transforming the results from the respective space back to the gamma-compressed domain. Table 4.11 summarizes the results, which showed two different things: (1) They demonstrated that modeling log-data or gamma-compressed data led to the best results, when measured as just discussed. Neither domain seemed to be clearly preferable, but both were substantially better than modeling linear-intensities. (2) The results once again showed how 5 × 5 models led to better image restoration results when compared to 3×3 models. In particular, the performance gap in this image inpainting application seemed to be even somewhat larger than what we found for image denoising.

4.6 Summary

In this chapter we studied the application of Fields of Experts to modeling natural images. We first studied the most important statistical properties of natural images and their impact on modeling images. Of particular importance are the long-range correlations between pixels, which as we showed are not fully captured by pairwise MRF models. The complex spatial structures in natural images strongly suggest the need for richer spatial models that go beyond using pairwise neighbors. Using Fields of Experts, we can successfully model such complex spatial structures as evidenced by the competitive application results shown here. In particular, we showed that FoEs considerably outperform pairwise MRFs and related techniques on both image denoising and image inpainting, while performing at a level that is very close to the state-of-the-art in each individual application (e. g., compared to wavelet methods in image denoising). This is despite the fact that FoEs are very generic, and can be applied (in a rather straightforward manner) to a broad range of applications not limited to the ones shown here.

In spite of the good performance, the model has a number of limitations, some of which will be discussed in more detail in Chapter 6. In particular, we saw that marginal statistics of natural images are not particularly well reproduced by the model. Nonetheless, even though pairwise MRFs better reproduce marginal derivative statistics, their inability to express complex spatial structures makes them less suitable in practical applications. This suggests that while marginal filter response statistics are still an important quality to reproduce with models of natural images, they are definitely not the only important aspect to consider.

We also presented a detailed analysis of various parameters and design decisions of the FoE model and the presented learning algorithm based on contrastive divergence. We found a number of interesting results in conjunction with image restoration applications, mostly image denoising. Let us recall the most important ones: (1) The filters learned with the FoE framework have high-frequency but are structured, both for Student t- and Charbonnier-experts; (2) Encouraging high-frequency filters through use of an appropriate basis improved performance; (3) 5×5 models with sufficiently many filters outperformed 3×3 models, but 7×7 FoEs did not seem to lead to additional performance gains; (4) Random filters did not perform as well as learned filters, although they performed relatively well in models with a limited clique size, if the basis in which they are drawn encouraged high-frequency filters; (5) Filters from patch-based models, such as Gabor-like filters found by ICA and PoE models, did not perform as well as the filters learned in conjunction with the FoE model. Unstructured looking filters, including minor components, were assigned the largest weights by the learning algorithm; (6) Regularizing the model through priors or by fixing certain parameters did not substantially improve results (or even deteriorated them); (7) Varying a number of other aspects of the learning algorithm did also not lead to substantially differing results, which demonstrated the robustness of the model. In conclusion, based on this analysis a reasonable model to use for many problems is a 5×5 FoE with 24 filters for which the filters are trained and initialized in “inverse whitened” space.

CHAPTER 5

Modeling Optical Flow

In this chapter we present an analysis of the spatial and temporal statistics of “natural” optical flow fields and a novel flow algorithm that exploits their spatial statistics. Training flow fields are constructed using range images of natural scenes and 3D camera motions recovered from hand-held and car-mounted video sequences. A detailed analysis of optical flow statistics in natural scenes is presented and learning methods are developed to learn a Markov random field model of optical flow. The prior probability of a flow field is modeled using the Field-of-Experts model developed in Chapter 3. This novel optical flow prior is compared with previous robust priors and is incorporated into a recent, accurate algorithm for dense optical flow computation. Experiments with natural and synthetic sequences illustrate how the learned optical flow prior quantitatively improves flow accuracy and how it captures the rich spatial structure found in natural scene motion.

5.1 Introduction

This chapter studies the statistics of optical flow in natural imagery and exploits high-order Markov random fields to obtain a rich probabilistic model for optical flow fields. This extends work on the analysis of image statistics in natural scenes and range images to the domain of image motion. In doing so we make connections to previous robust statistical formulations of optical flow smoothness priors. We furthermore apply the developed Field-of-Experts framework to the task of modeling the a-priori (spatial) statistics of optical flow. We extend a recent (and very accurate) optical flow method [Bruhn et al., 2005] with this new prior and provide an algorithm for estimating optical flow from pairs of images. We quantitatively compare the learned prior with more traditional robust priors and find that in our experiments the accuracy is improved by about 10% while removing the need for tuning the scale parameter of the traditional priors.

As discussed in Chapter 4, natural image statistics have received intensive study [Ruderman,

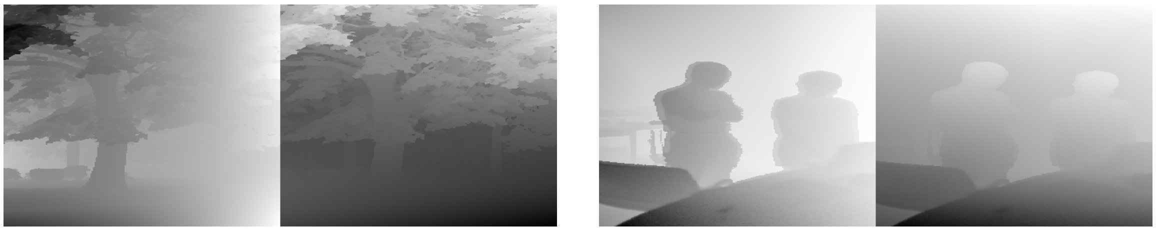


Figure 5.1: **Flow fields generated for an outdoor (left) and an indoor (right) scene.** The horizontal motion $\mathbf{u}$ is shown on the left, the vertical motion $\mathbf{v}$ on the right of each figure; dark/light means negative/positive motion (independently scaled to $0 \ldots 255$ for display).

1994; Huang, 2000; Lee et al., 2001], but the spatial and temporal statistics of optical flow are relatively unexplored because databases of natural scene motions are currently unavailable. One of the contributions of this dissertation is the development of such a database.

The spatial statistics of the motion field (i. e., the ideal optical flow) are determined by the interaction of (1) camera motion; (2) scene depth; and (3) the independent motion of objects. Here we focus on rigid scenes and leave independent motion for future work (though we believe the statistics from rigid scenes are useful for scenes with independent motion and will show experimental results with independent motion). To generate a realistic database of optical flow fields we exploit the Brown range image database [Lee and Huang, 2000], which contains depth images of complex scenes including forests, indoor environments, and generic street scenes. Given 3D camera motions and range images we generate flow fields that have the rich spatial statistics of "natural" flow fields[1]. A set of natural 3D motions was obtained from both hand-held and car-mounted cameras performing a variety of motions including translation, rotation, and active fixation. The 3D motion was recovered from these video sequences using commercial software [2d3 Ltd., 2002]. Figure 5.1 shows two example flow fields generated using the 3D motions and the range images.

This study shows that the first-derivative statistics of optical flow fields are very heavy-tailed as are the statistics of natural images. We observe that the first derivative statistics are well modeled by heavy tailed distributions such as the Student t-distribution. This provides a connection to previous robust statistical methods for recovering optical flow that modeled spatial smoothness using robust functions [Black and Anandan, 1996] and suggests that the success of robust methods is due to the fact that they capture the first-order spatial statistics of optical flow.

Our goal here is to go beyond such local (first derivative) models and formulate a Markov

---

[1] By "natural" here we primarily mean spatially structured and relevant to humans. Consequently the data represents scenes humans inhabit and motions they perform. We do not include nonrigid or textural motions resulting from independent movement in the environment (though these are also "natural"). Motions of a person's body and motions they cause in other objects are also excluded at this time.

random field (MRF) prior that captures richer spatial statistics present in larger neighborhoods. To that end, we exploit the Field-of-Experts model developed in the preceding parts of this dissertation. Just like in the application of the FoE to modeling natural images, we model the clique potentials in terms of various linear filter responses, but now of the image *motion*. Like for images, we use Student t-distributions for the experts, and we learn both the parameters of the distribution and the filters themselves using contrastive divergence.

We compute optical flow using the learned prior as a smoothness term. The log-prior is combined with a data term and we minimize the resulting energy (negative log-posterior). While the exact choice of data term is not relevant for the analysis, here we use the recent approach of Bruhn et al. [2005], which replaces the standard optical flow constraint equation with a tensor that integrates brightness constancy constraints over a spatial neighborhood. We present an algorithm for estimating dense optical flow and compare the performance of standard robust spatial terms with the learned FoE model on both synthetic and natural imagery.

5.1.1 Previous work

There has been a great deal of work on modeling natural image statistics [Ruderman, 1994; Olshausen and Field, 1996; Huang, 2000; Lee et al., 2001; Srivastava et al., 2003] facilitated by the existence of large image databases. One might expect optical flow statistics to differ from image statistics in that there is no equivalent of “surface markings” in motion and all structure in rigid scenes results from the shape of surfaces and the discontinuities between them. In this way it seems plausible that flow statistics share more with depth statistics. Unlike optical flow, direct range sensors exist and a time-of-flight laser was used in [Huang et al., 2000] to capture the depth in a variety of scenes including residential street scenes, forests, and indoor environments. Scene depth statistics alone, however, are not sufficient to model optical flow, because image motion results from the combination of the camera motion and depth. While models of self-motion in humans and animals [Lewen et al., 2001; Betsch et al., 2004] have been studied, we are unaware of attempts to learn or exploit a database of camera motions captured by a moving camera in natural scenes.

The most similar work to ours also uses the Brown range image database to generate realistic synthetic flow fields [Calow et al., 2004]. The authors use a gaze tracker to record how people view the range images and then simulate their motion into the scene with varying fixation points. Their focus is on human perception of flow and consequently they analyze a retinal projection of the flow field. They also limit their analysis to first-order spatial statistics and do not propose an algorithm for exploiting these statistics in the computation of optical flow.

Previous work on learning statistical models of video focuses on the statistics of the changing brightness patterns rather than the flow it gives rise to. For example, adopting a classic sparse-coding hypothesis, video sequences can be represented using a set of learned spatio-temporal filters [van Hateren and Ruderman, 1998]. Other work has focused on the statistics of the classic brightness constancy assumption (and how it is violated) rather than the spatial statistics of the flow field [Simoncelli et al., 1991; Fermüller et al., 2001].

The lack of training data has limited research on learning spatial models of optical flow. One exception is the work by Fleet et al. [2000] in which the authors learn local models of optical flow from examples using principal component analysis (PCA). In particular, they use synthetic models of moving occlusion boundaries and bars to learn linear models of the flow for these motion features. Local, non-overlapping models such as these may be combined in a spatio-temporal Bayesian network to estimate coherent global flow fields [Fleet et al., 2002]. While promising, these models cover only a limited range of the variation in natural flow fields.

There is related interest in the statistics of optical flow in the video retrieval community; for example, Fablet and Bouthemy [2001] learn statistical models using a variety of motion cues to classify videos based on their spatio-temporal statistics. These methods, however, do not focus on the estimation of optical flow.

The formulation of smoothness constraints for optical flow estimation has a long history [Horn and Schunck, 1981], as has its Bayesian formulation in terms of Markov random fields [Murray and Buxton, 1987; Marroquin et al., 1987; Konrad and Dubois, 1988; Black and Anandan, 1991; Heitz and Bouthemy, 1993]. Such formulations of optical flow estimate the flow using Bayesian inference based on a suitable posterior distribution. Given an image sequence $\mathbf{f}$, the horizontal flow $\mathbf{u}$ and the vertical flow $\mathbf{v}$ are typically found by maximizing the posterior distribution $p(\mathbf{u}, \mathbf{v} \,|\, \mathbf{f})$ with respect to $\mathbf{u}$ and $\mathbf{v}$. Other inference methods estimate the flow using sampling [Barbu and Yuille, 2004] or by computing expectations. As in the general case discussed before, the posterior can be broken down using Bayes' rule into a likelihood term $p(\mathbf{f} \,|\, \mathbf{u}, \mathbf{v})$, also called the data term, and an optical flow prior $p(\mathbf{u}, \mathbf{v})$, also called the spatial term:

$$\arg\max_{\mathbf{u},\mathbf{v}} \; p(\mathbf{u}, \mathbf{v} \,|\, \mathbf{f}) = \arg\max_{\mathbf{u},\mathbf{v}} \; p(\mathbf{f} \,|\, \mathbf{u}, \mathbf{v}) \cdot p(\mathbf{u}, \mathbf{v}). \tag{5.1}$$

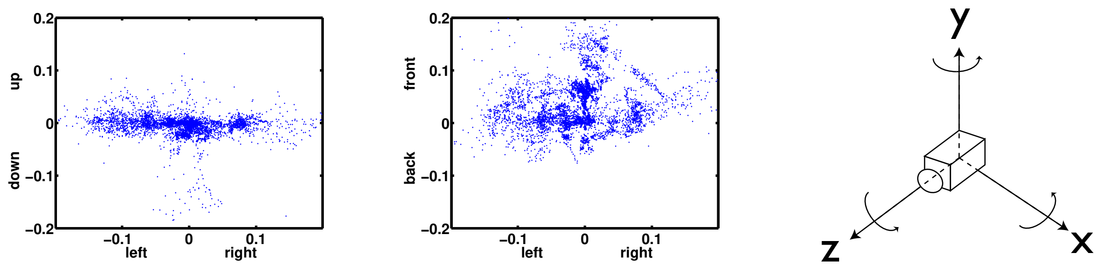
The data term enforces the brightness constancy assumption, which underlies most flow estimation techniques [e. g., Horn and Schunck, 1981]; the spatial term enforces spatial smoothness for example using a Markov random field model. Previous work, however, has focused on very local prior models that are typically formulated in terms of the first differences in the optical flow (i. e., the nearest neighbor differences). This can model piecewise constant or smooth flow but not more complex spatial structures. A large class of optical flow techniques make use of spatial regularization terms derived from the point of view of variational methods, PDEs, and nonlinear diffusion [Proesmans et al., 1994; Alvarez et al., 2000; Weickert and Schnörr, 2001; Bruhn et al., 2005; Scharr and Spies, 2005; Cremers and Soatto, 2005; Papenberg et al., 2006; Ben-Ari and Sochen, 2006]. These methods also exploit only local measures of the flow gradient and are not directly motivated by statistical properties of natural flow fields. Other work has imposed geometric rather than spatial smoothness constraints on multi-frame optical flow [Irani, 1999].

Weiss and Adelson [1998] propose a Bayesian model of motion estimation to explain human perception of visual stimuli. In particular, they argue that the appropriate prior prefers "slow and smooth" motion (see also [Lu and Yuille, 2006]). Their stimuli, however, are too simplistic to probe the nature of flow priors in complex scenes. We find these statistics are more like those of natural images in that the motions are piecewise smooth; large discontinuities give rise to heavy tails in the first derivative statistics. Our analysis suggests that a more appropriate flow prior is "mostly slow and smooth, but sometimes fast and discontinuous".

5.2 Spatial Statistics of Optical Flow

5.2.1 Obtaining training data

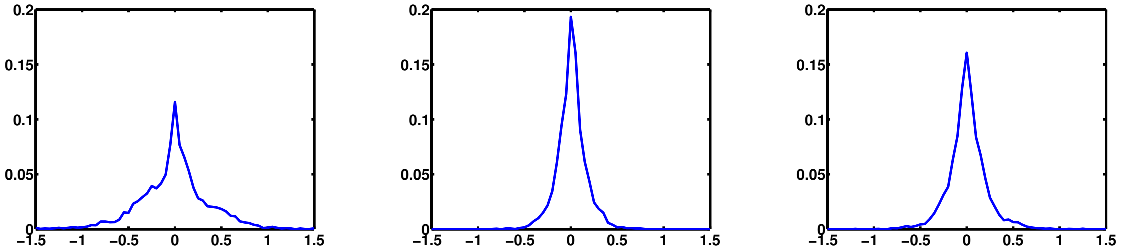
One of the key challenges in analyzing and learning the spatial statistics of optical flow is to obtain suitable optical flow data. The issue here is that optical flow cannot be directly measured, which makes the statistics of optical flow a largely unexplored field. Synthesizing realistic flow fields is thus the only viable option for studying, as well as learning the statistics of optical flow. Optical flow has previously been generated using computer graphics techniques in order to obtain benchmarks with ground truth, for example in case of the Yosemite sequence [Barron et al., 1994]. In most cases, the underlying scenes are very simple, however. Our goal here is to create a database of realistic optical flow fields as they arise in natural as well as man-made scenes. It is unlikely that the rich statistics will be captured by any manual construction of the training data as in [Fleet et al., 2000], which uses simple polygonal object models. We also cannot rely on optical flow as computed from image sequences, because then we would learn the statistics of the algorithm used to compute the flow. It would be possible to use complex scene models from computer graphics, but such scenes would have to be very complex to capture the structure of real scenes. Instead we rely on range images from the Brown range image database [Lee and Huang, 2000], which



(a) Left-right (x) and up-down (y) translational motion.

(b) Left-right (x) and forward-backward (z) translational motion.

(c) Camera coordinate system.



(d) Yaw (rotation around y).

(e) Pitch (rotation around x).

(f) Roll (rotation around z).

Figure 5.2: **Statistics of camera motion in our database.** (a),(b) Scatter plots of the translational camera motion between subsequent frames (scale in meters). (d)–(f) Histograms of camera rotation between subsequent frames (angle in degrees).

provides accurate scene depth information for a set of 197 indoor and outdoor scenes (see Fig. 5.3). Optical flow has been generated from range images by Calow et al. [2004], but they focus on a synthetic model of human gaze and ego-motion. Unlike Calow et al. [2004], we focus on optical flow fields as they arise in machine vision applications as opposed to human vision[2].

The range image database we use captures information about surfaces and surface boundaries in natural scenes, but it is completely static. Hence we focus only on the case of ego-motion in static scenes. A rigorous study of the optical flow statistics of independently moving objects will remain the subject of future work. Despite this limitation, the range of motions represented is broad and varied.

Camera motion. Apart from choosing appropriate 3D scenes, finding suitable camera motion data is another challenge. In order to cover a broad range of possible frame-to-frame camera motions, we used a database of 67 video clips of approximately 100 frames, each of which was shot using a hand-held or car-mounted video camera. The database is comprised of various kinds of motion, including forward walking and moving the camera

---

[2] It is worth noting that our goal is to learn the prior probability of the motion field (i. e., the projected scene flow) rather than the apparent motion (i. e., the optical flow) since most applications of optical flow estimation seek this "true" motion field.

Figure 5.3: **Example range scenes from the Brown range image database [Lee and Huang, 2000].** Intensity here codes for depth with distant objects appearing brighter. Regions for which no range estimate could be obtained are shown in black.

around an object of interest. The extrinsic and intrinsic camera parameters were recovered using the *boujou* software system [2d3 Ltd., 2002] from a number of tracked feature points; the underlying camera model is a simple perspective pinhole camera. Figure 5.2 shows empirical distributions of the camera translations and rotations in the database. The plots reveal that left-right movements are more common than up-down movements and that moving into the scene occurs more frequently than moving out of the scene. Similarly the empirical distributions of camera rotation show that yaw occurs more frequently than pitch and roll motions of the camera.

Synthesizing optical flow. To generate optical flow from range and camera motion data we use the following procedure: Given a range scene, we build a triangular mesh from the depth measurements, which allows us to view the scene from any view point. However, only viewpoints near the location of the range scanner will lead to the intended appearance. Given two camera transformations of subsequent frames, rays are then cast through every pixel of the first frame to find the corresponding scene point from the range image. Each of these scene points is then projected onto the image plane of the second frame. The optical flow is simply given by the difference in image coordinates under which a scene point is viewed in each of the two cameras. We used this procedure to generate a database of 800 optical flow fields, each 250×200 pixels large[3]. Figure 5.1 shows example flow fields from this

---

[3] The database is available at `http://www.cs.brown.edu/~roth/research/flow/downloads.html`.

database. Note that we do not explicitly represent the regions of occlusion or disocclusion in the database. While occlusions for a particular camera motion can be detected based on the polygonal scene model, they are not explicitly modeled in our spatial model of optical flow. While our learned MRF model will capture motion boundaries implicitly, a rigorous treatment of occlusions [cf., Fleet et al., 2002; Ross and Kaelbling, 2005] will remain future work.

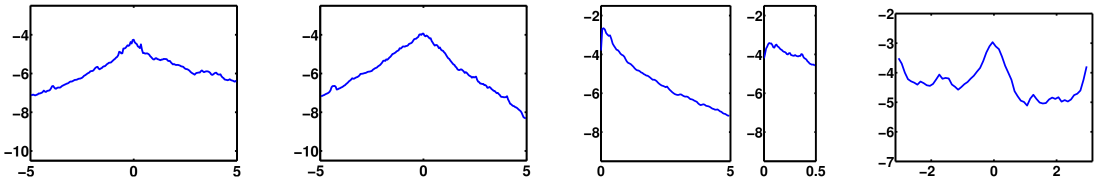
Additionally, we have to select appropriate combinations of range scenes and 3D camera motions. In other work [Roth and Black, 2005b], we independently sampled range scenes as well as camera motions. This is a simplifying assumption, however, because the kind of camera motion executed may depend on the kind of scene that is viewed. In particular, the amplitude of the camera motion may be dependent on the scale of the viewed scene. It seems likely, for example, that camera motion tends to be slower when objects are nearby, than when very distant objects are viewed. To account for this potential dependency, we go beyond this and introduce a coupling between camera motion and scene structure. We first sample a random camera motion and compute a depth histogram of the feature points being tracked in the original sequence used to recover the camera motion. These feature points are provided by the *boujou* system and give a coarse description of the scene being filmed. We then compute depth histograms for 4925 potential views of various range scenes (25 different views of each scene), and find the Bhattacharyya distance [Kailath, 1967]

$$d(\mathbf{p}, \mathbf{q}^{(j)}) = \sqrt{1 - \sum_i \sqrt{p_i \cdot q_i^{(j)}}} \tag{5.2}$$

between the depth histogram of the scene corresponding to the camera motion ($\mathbf{p}$) and each of the candidate views of the range scenes ($\mathbf{q}^{(j)}$). We define a weight $w_j := (1 - d(\mathbf{p}, \mathbf{q}^{(j)}))^2$ for each candidate range scene, which assigns greater weight to range scenes with similar depth statistics to the camera motion scene. A range scene view is finally sampled according to these weights, which achieves a coupling of the depth and camera motion statistics. In practice we found that many of the range scenes have very distant objects compared to the scenes for which we have camera motion. We found that if the range scenes are scaled down using a global scaling factor of 0.25, then the range statistics much better matched those for the scenes used to capture the camera motion. We hence used this scaling factor when sampling scene/motion pairs.

5.2.2 Velocity statistics

Using this database, we are able to study several statistical properties of optical flow. Figure 5.4 shows log-histograms of the image velocities in various forms. In addition to the

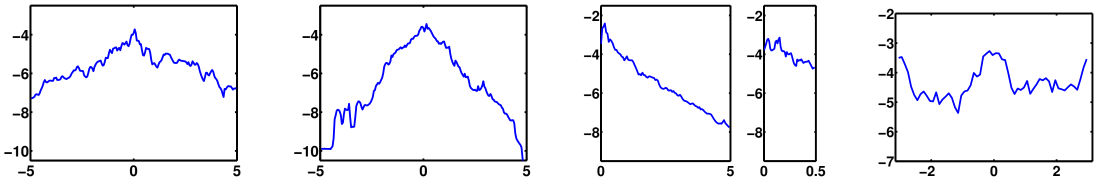


(a) Horizontal velocity u.

(b) Vertical velocity v.

(c) Velocity r.

(d) Orientation θ.

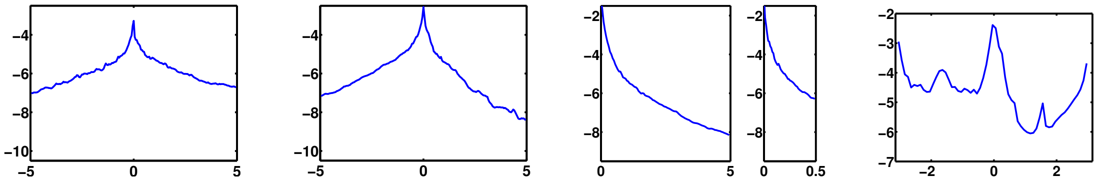


(e) Horizontal velocity u, only rotational motion.

(f) Vertical velocity v, only rotational motion.

(g) Velocity r, only rotational motion.

(h) Orientation θ, only rotational motion.



(i) Horizontal velocity u, only translational motion.

(j) Vertical velocity v, only translational motion.

(k) Velocity r, only translational motion.

(l) Orientation θ, only translational motion.

Figure 5.4: **Velocity and orientation histograms of the optical flow in our database (log scale).** The right plot in parts (c), (g), and (k) shows details of the left plot.

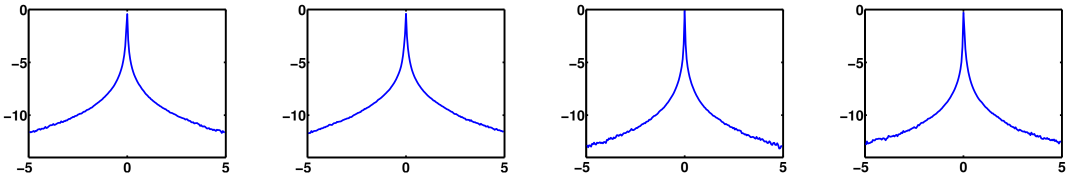
optical flow database described in the previous section, we also study two further flow databases: one resulting from purely translational camera motion and one from purely rotational camera motion. These allow us to attribute various properties to each type of motion. We observe that the vertical velocity in Figure 5.4(b) is roughly distributed like a Laplacian distribution; the horizontal velocity in Figure 5.4(a) shows a slightly broader histogram that falls off less quickly. This is consistent with our observations that horizontal camera motions are more common. If we factor the motion into rotational and translational components, we find that the heavy tails of the velocity histograms are due to the translational camera motion (Figs. 5.4(i),(j)), while the image velocity from rotational camera motion covers a smaller range (Figs. 5.4(e),(f)). Maybe somewhat surprisingly the vertical image motion due to translational camera motion in Figure 5.4(j) appears to be biased toward negative values, which correspond to motion toward the bottom of the image. This nevertheless has a quite simple explanation: From the camera motion statistics we know that moving in the viewing direction occurs quite frequently; for hand-held and car-mounted cameras this motion will typically be along the ground plane. Finally, the part of the image below the horizon typically occupies more than half of the whole image, which causes the focus of expansion to be in the top half of the image. Because of that, downward image motion dominates upward image motion for this common type of camera motion.

Figure 5.4(c) shows the magnitude of the velocity, which falls off in a manner similar to a Laplacian distribution [Simoncelli et al., 1991]. The statistics of natural image motion hence suggest that image motions are typically slow. The results of Weiss and Adelson [1998] suggest that humans may exploit this prior information in their estimation of image motion. The heavy-tailed nature of the velocity histograms indicates that while slow motions are typical, this assumption is still violated fairly frequently for “natural” optical flow. From the histograms we can also see that very small motions (near zero) seem to occur slightly less frequently that somewhat larger motions, which is mainly attributable to rotational camera motion (Fig. 5.4(g)). This suggests that the camera is rarely totally still and is often rotated at least a small amount. The orientation histogram in Figure 5.4(d) again shows the preference for horizontal motion (the bumps at 0 and π). However, there are also smaller spikes indicating somewhat frequent up-down motion (at $\pi/2$); we observe similar properties for rotational (Fig. 5.4(h)) and translational camera motion (Fig. 5.4(l)).

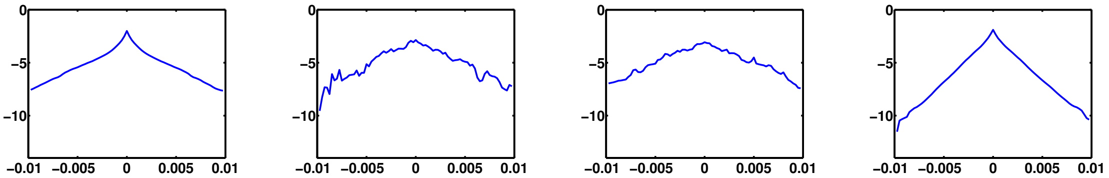
5.2.3 Derivative statistics

Figure 5.5 shows the first derivative histograms of the spatial derivatives for both horizontal and vertical image motion. For flow from translational and combined translational/rotational camera motion, the distributions are all heavy-tailed and strongly resemble Student t-distributions (Figs. 5.5(a)–(d) and 5.5(i)–(l)). Such distributions have also been encountered in the study of natural images, [e. g., Huang, 2000; Lee et al., 2001; Grenander and Srivastava, 2001]. In natural images, the image intensity is often locally smooth, but occasionally shows large jumps at object boundaries or in fine textures, which give rise to substantial probability mass in the tails of the distribution. Furthermore, the study of range images [Huang et al., 2000] has shown similar derivative statistics. For scene depth the heavy-tailed distributions arise from depth discontinuities mostly at object boundaries. Because the image motion from camera translation is directly dependent on the scene depth, it is not surprising to see similar distributions for optical flow. Optical flow from purely rotational camera motion on the other hand is independent of scene depth, which is reflected in the much less heavy-tailed derivative histograms (Figs. 5.5(e)–(h)).

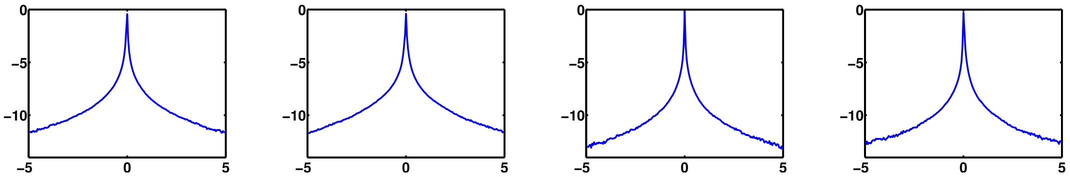
The large peaks at zero of the derivative histograms show that optical flow is typically very smooth. Aside from prior information on the flow magnitude, the work by Weiss and Adelson [1998] suggested that humans also use prior information about the smoothness of optical flow. The heavy-tailed nature of the statistics of natural optical



(a) $\partial u/\partial x$. (b) $\partial u/\partial y$. (c) $\partial v/\partial x$. (d) $\partial v/\partial y$.



(e) $\partial u/\partial x$, only rotational motion. (f) $\partial u/\partial y$, only rotational motion. (g) $\partial v/\partial x$, only rotational motion. (h) $\partial v/\partial y$, only rotational motion.



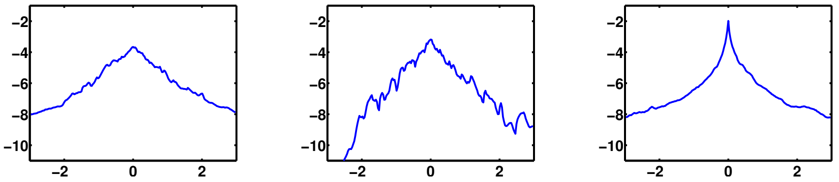
(i) $\partial u/\partial x$, only translational motion. (j) $\partial u/\partial y$, only translational motion. (k) $\partial v/\partial x$, only translational motion. (l) $\partial v/\partial y$, only translational motion.

Figure 5.5: **Spatial derivative histograms of optical flow in our database (log scale).** $\partial u/\partial x$, for example, denotes the horizontal derivative of the horizontal flow.

flow suggests that flow discontinuities still occur with considerable frequency. Hence an appropriate prior would be "mostly slow" and "mostly smooth". This suggests a direction for further study in human vision to see if such priors are used.

Furthermore, the observed derivative statistics likely explain the success of robust statistical formulations for optical flow computation based on M-estimators [e. g., Black and Anandan, 1991]. In [Black and Anandan, 1991] the spatial smoothness term was formulated as a robust function (Lorentzian) of the horizontal and vertical partial derivatives of the flow field. This robust function fits the marginal statistics of these partial derivatives very well.

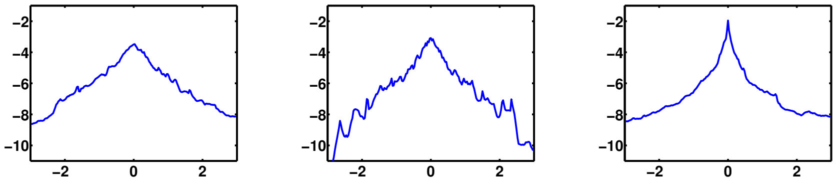
Temporal statistics. Even though this chapter is mainly concerned with analyzing and modeling the spatial statistics of optical flow, we also performed an analysis of the temporal properties of the flow. In particular, we generated pairs of flow fields from three subsequent camera viewpoints, and computed the difference of the optical flow between subsequent pairs in the image coordinate system. Figure 5.6 shows the statistics of this temporal derivative, again for all three types of camera motion. We find that similar to the spatial derivatives, the temporal derivatives of flow from translational and combined translational/rotational camera motion are more heavy-tailed (Figs. 5.6(a),(d),(c),(f)) than for flow from purely



(a) $\partial u/\partial t$.

(b) $\partial u/\partial t$, only rotational motion.

(c) $\partial u/\partial t$, only translational motion.

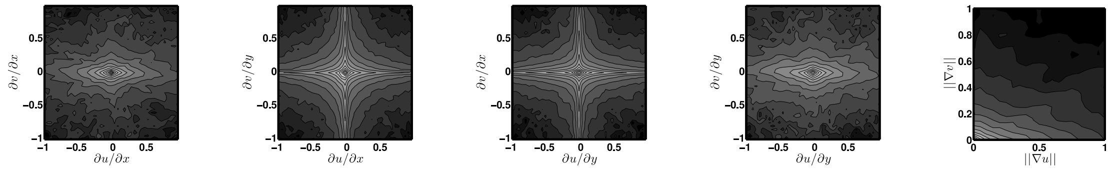


(d) $\partial v/\partial t$.

(e) $\partial v/\partial t$, only rotational motion.

(f) $\partial v/\partial t$, only translational motion.

Figure 5.6: **Temporal derivative histograms of optical flow in our database (log scale).**



(a) $\partial u/\partial x$ and $\partial v/\partial x$ (MI 0.11 bits)

(b) $\partial u/\partial x$ and $\partial v/\partial y$ (MI 0.06 bits)

(c) $\partial u/\partial y$ and $\partial v/\partial x$ (MI 0.03 bits)

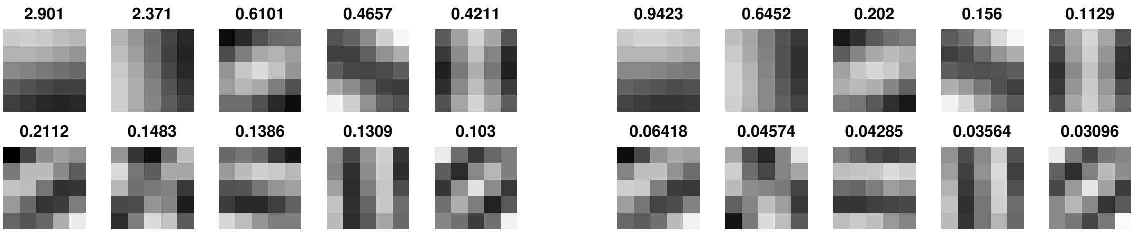
(d) $\partial u/\partial y$ and $\partial v/\partial y$ (MI 0.13 bits)

(e) $||\nabla u||$ and $||\nabla v||$ (MI 0.12 bits)

Figure 5.7: **Joint log-histograms of derivatives of horizontal and vertical flow and their mutual information (MI).** The mutual information expresses how much information (in bits) one derivative value contains about the other; the small values here indicate approximate statistical independence.

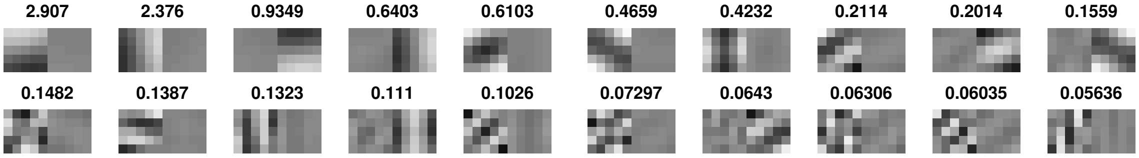
rotational motion (Figs. 5.6(b),(e)). In the remainder of this work, we do not pursue the temporal statistics any further, but the extension of the model proposed in Section 5.3 to the temporal domain will be an interesting avenue for future work.

Joint statistics. We also obtained joint empirical histograms of derivatives of the horizontal and vertical flow. Figure 5.7 shows log-histograms of various derivatives, where a derivative of the horizontal flow is plotted against a derivative of the vertical flow. We can see from the shape of the empirical histograms that the derivatives of horizontal and vertical flow are largely independent. This observation is reinforced quantitatively when considering the mutual information (MI) between the various derivatives. We estimated the mutual information directly from the empirical histograms. Figure 5.7 gives the MI values for all considered joint histograms.



(a) Horizontal velocity $\mathbf{u}$.

(b) Vertical velocity $\mathbf{v}$.



(c) Joint horizontal velocity $\mathbf{u}$ & vertical velocity $\mathbf{v}$.

Figure 5.8: **First 10 principal components** (20 for $\mathbf{u}$ & $\mathbf{v}$) of the image velocities in 5×5 patches. The numbers denote the variance accounted for by the principal component.

5.2.4 Principal component analysis

We also performed principal component analysis on small patches of flow of various sizes. Figure 5.8 shows the results for horizontal flow $\mathbf{u}$ and vertical flow $\mathbf{v}$ in 5×5 patches. The principal components of horizontal and vertical flow look very much alike, but the variance of the vertical flow components is smaller due to the observed preference for horizontal motion. We can see that a large portion of the flow variance is focused on the first few principal components which, as with images (cf. Fig. 4.5), resemble derivative filters of various orders [cf., Fleet et al., 2000]. A joint analysis of horizontal and vertical flow as shown in Figure 5.8(c) reveals that the principal components treat horizontal and vertical flow largely independently, i. e., one of the components is constant while the other is not.

5.3 Modeling Optical Flow

To capture the spatial statistics of optical flow we use the Fields-of-Experts model as developed in Chapter 3. This allows us to model the prior probability of optical flow fields using a high-order Markov random field, which is furthermore learned from data. We argue that spatial regularization of optical flow will benefit from prior models that capture interactions beyond adjacent pixels. While as discussed before, Markov random fields of high-order have been shown to be quite powerful for certain low-level vision applications, to our knowledge learned high-order MRF models have not been applied to the problem of optical flow estimation before.

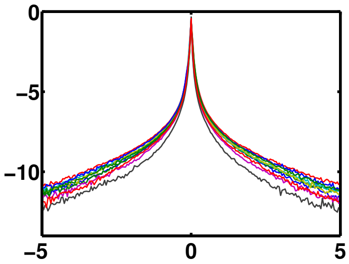


Figure 5.9: **Log-histograms of filter responses of 10 zero-mean, unit-variance random filters** applied to the horizontal flow in our optical flow database.

In Section 5.2.3 we have seen that the derivatives of horizontal and vertical motion are largely independent, hence for simplicity, we will typically treat horizontal and vertical image motions separately, and learn two independent models. Hence, the joint probability of the flow field $p(\mathbf{u}, \mathbf{v})$ is simply the product of the probabilities of the two components $p(\mathbf{u}) \cdot p(\mathbf{v})$. Each flow field component $\mathbf{x} \in \{\mathbf{u}, \mathbf{v}\}$ is modeled as a Field of Experts (cf. Eq. (3.3)):

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \alpha_i). \tag{5.3}$$

As we will discuss in more detail below, most typical prior models for optical flow can actually be expressed in this more general MRF framework.

In our experiments we have observed that responses of linear filters applied to flow components in the optical flow database show histograms that are typically well fit by t-distributions. Figure 5.9 illustrates this with the response statistics of 10 zero-mean random filters on our optical flow database. Motivated by this observation, we choose Student t-distributions with parameter α_i as the expert functions, just as for natural images (cf. Eq. (3.10)):

$$\phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \alpha_i) = \left(1 + \frac{1}{2}(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)})^2\right)^{-\alpha_i}. \tag{5.4}$$

We should note here that this model can easily be extended to a joint model of horizontal and vertical flow by considering cliques of size $m \times m \times 2$. The mathematical form of such a model is essentially the same, except that the random vector $\mathbf{x}$ is arranged on a three-dimensional lattice.

Training is done using contrastive divergence as described in Section 3.3. For our experiments, we trained various FoE models: 3×3 cliques with 8 filters, 5×5 cliques with 24 filters, as well as a joint $3 \times 3 \times 2$ model with 16 filters. As for images, we restrict the filters so that they do not capture the mean velocity in a patch and are thus only sensitive to relative motion. We train the filters in a transformed space based on “inverse” whitening as before. The training data consisted of 2000 flow fields of size 15×15. These small flow fields were selected and cropped uniformly at random from the synthesized flow fields described in Section 5.2. The filters were initialized randomly by drawing each coefficient in the transformed space from a unit normal distribution; the expert parameters were initialized uniformly with $\alpha_i = 1$.

In the context of optical flow estimation, the prior knowledge about the flow field is typically expressed in terms of an energy function $E(\mathbf{x}) = -\log p(\mathbf{x})$. Accordingly, we can express the energy for the FoE prior model as (cf. Eq. (3.5))

$$E_{\text{FoE}}(\mathbf{x}) = -\sum_{k=1}^{K}\sum_{i=1}^{N} \log \phi(\mathbf{J}_i^{\text{T}}\mathbf{x}_{(k)}; \alpha_i) + \log Z. \tag{5.5}$$

Note that for fixed parameters α and $\mathbf{J}$, the partition function Z is constant, and can thus be ignored when estimating flow.

The optical flow estimation algorithm we propose in the next section relies on the gradient of the energy function with respect to the flow field. We thus use the gradient expression from Eq. (3.24) to compute the gradient.

5.3.1 Comparison to traditional regularization approaches

Many traditional regularization approaches for optical flow can be formalized in a very similar way, and some of them are in fact restricted versions of the more general FoE model introduced above. One widely used regularization technique is based on the gradient magnitude of the flow field components [Bruhn et al., 2005]. Because the gradient magnitude is typically computed using finite differences in horizontal and vertical directions, the maximal cliques of the corresponding MRF model consist of 4 pixels arranged in a diamond-like shape. Assuming that the flow field component $\mathbf{x} \in \{\mathbf{u}, \mathbf{v}\}$ is indexed as $x_{i,j}$, the model can be formalized as

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{Z}\prod_{i,j} \psi(x_{i+1,j}, x_{i-1,j}, x_{i,j+1}, x_{i,j-1}) \\ &= \frac{1}{Z}\prod_{i,j} \hat{\psi}\left(\sqrt{(x_{i+1,j} - x_{i-1,j})^2 + (x_{i,j+1} - x_{i,j-1})^2}\right). \end{aligned} \tag{5.6}$$

If $\hat{\psi}$ is chosen to be a Gaussian distribution, one obtains a standard linear regularizer; heavy-tailed potentials on the other hand will lead to robust, nonlinear regularization [Bruhn et al., 2005]. Because computing the gradient magnitude involves a nonlinear transformation of the pixel values, this model cannot be directly mapped into the FoE framework introduced above.

A second regularization approach that has been used frequently throughout the literature is based on component derivatives directly as opposed to the gradient magnitude [e. g., Horn and Schunck, 1981; Black and Anandan, 1996]. Assuming the same notation as above, this prior model can be formalized as

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{Z} \prod_{i,j} \psi(x_{i+1,j}, x_{i,j}) \cdot \prod_{i,j} \psi(x_{i,j+1}, x_{i,j}) \\ &= \frac{1}{Z} \prod_{i,j} \hat{\psi}(x_{i+1,j} - x_{i,j}) \cdot \prod_{i,j} \hat{\psi}(x_{i,j+1} - x_{i,j}). \end{aligned} \tag{5.7}$$

As above, $\hat{\psi}$ can be based on Gaussian or robust potentials and parametrized accordingly. In contrast to the gradient magnitude prior, the potentials of this prior are based on a linear transformation of the pixel values in a clique. Because of that, it is a special case of the FoE model in Eq. (5.3); here, the filters are simple derivative filters based on finite differences. While the type of the filter, i. e., the direction of the corresponding filter vector $\mathbf{J}_i$, is fixed in this simple pairwise case, the norm of the filters $\mathbf{J}_i$ is not fixed and neither are the expert parameters α_i. As for the general FoE model, these have to be chosen appropriately, which we can do by learning them from data. As before, we do this based on approximate maximum likelihood estimation using contrastive divergence learning.

5.4 Optical Flow Estimation

In order to demonstrate the benefits of learning the spatial statistics of optical flow, we integrate our model with a recent, competitive optical flow method and quantitatively compare the results. As a baseline algorithm we chose the combined local-global method (CLG) as proposed by Bruhn et al. [2005]. Global methods typically estimate the horizontal and vertical image velocities $\mathbf{u}$ and $\mathbf{v}$ by minimizing an energy functional of the form [Horn and Schunck, 1981]

$$E(\mathbf{u}, \mathbf{v}) = \int_I \rho_\mathrm{D}(I_x \mathbf{u} + I_y \mathbf{v} + I_t) + \lambda \cdot \rho_\mathrm{S}\left(\sqrt{|\nabla \mathbf{u}|^2 + |\nabla \mathbf{v}|^2}\right) \,\mathrm{d}x\,\mathrm{d}y. \tag{5.8}$$

ρ_D and ρ_S are robust penalty functions, such as the Lorentzian [Black and Anandan, 1996]; I_x, I_y, I_t denote the spatial and temporal derivatives of the image sequence. Note that this is a special case of the general variational framework from Eq. (2.9). The first term in Eq. (5.8) is the so-called data term that enforces the brightness constancy assumption (here written as the first-order optical flow constraint). The second term is the so-called spatial term, which enforces (piecewise) spatial smoothness. Since in this model, the data term relies on a local linearization of the brightness constancy assumption, such methods are usually used in a coarse-to-fine fashion [e. g., Black and Anandan, 1996], which allows the estimation of large displacements. For the remainder, we will assume that large image velocities are handled using such a coarse-to-fine scheme with appropriate image warping (see also Appendix B.2).

The combined local-global method extends this framework through local averaging of the brightness constancy constraint by means of a structure tensor. This connects the method to local optical flow approaches that spatially integrate image derivatives to estimate the image velocity locally [Lucas and Kanade, 1981]. Using $\nabla I = (I_x, I_y, I_t)^{\mathrm{T}}$ we can define a spatio-temporal structure tensor as $\mathbf{K}_\sigma(I) = G_\sigma * \nabla I \nabla I^{\mathrm{T}}$, where G_σ denotes a Gaussian convolution kernel with width σ. To compute the image derivatives, we rely on optimized $4 \times 4 \times 2$ filters as developed by Scharr [2004], unless otherwise remarked. The CLG approach estimates the optical flow by minimizing the energy functional

$$E_{\mathrm{CLG}}(\mathbf{w}) = \int_I \rho_{\mathrm{D}} \left(\sqrt{\mathbf{w}^{\mathrm{T}} \mathbf{K}_\sigma(I) \mathbf{w}} \right) + \lambda \cdot \rho_{\mathrm{S}} \left(|\nabla \mathbf{w}| \right) \, \mathrm{d}x \, \mathrm{d}y, \tag{5.9}$$

where $\mathbf{w} = (\mathbf{u}, \mathbf{v}, 1)^{\mathrm{T}}$. In the following we will only work with spatially discrete flow representations. Consequently we assume that the data term of $E_{\mathrm{CLG}}(\mathbf{w})$ is given in discrete form $E_{\mathrm{D}}(\mathbf{w})$, where $\mathbf{w}$ is now a vector of all horizontal and vertical velocities in the image (see Appendix B.1 for details). Experimentally, the CLG approach has been shown to be one of the best currently available optical flow estimation techniques. The focus of this chapter is the spatial statistics of optical flow; hence, we will only make use of the 2D-CLG approach, i. e., only two adjacent frames will be used for flow estimation.

We refine the CLG approach by using a spatial regularizer that is based on the learned spatial statistics of optical flow. Many global optical flow techniques enforce spatial regularity or "smoothness" by penalizing large spatial gradients. In Section 5.3 we have discussed how we can learn high-order Markov random field models of optical flow, which we use here as a spatial regularizer for flow estimation. Our objective is to minimize the energy

$$E(\mathbf{w}) = E_{\mathrm{D}}(\mathbf{w}) + \lambda \cdot E_{\mathrm{FoE}}(\mathbf{w}). \tag{5.10}$$

Since $E_{\mathrm{FoE}}(\mathbf{w})$ is non-convex, minimizing Eq. (5.10) is generally difficult. Depending on the choice of the robust penalty function, the data term may in fact be non-convex, too. We will not attempt to find the global optimum of the energy function, but instead perform a simple local optimization. At any local extremum of the energy it holds that

$$\mathbf{0} = \nabla_{\mathbf{w}} E(\mathbf{w}) = \nabla_{\mathbf{w}} E_{\mathrm{D}}(\mathbf{w}) + \lambda \cdot \nabla_{\mathbf{w}} E_{\mathrm{FoE}}(\mathbf{w}). \tag{5.11}$$

Method with spatial and data terms	mean AAE	AAE std. dev.
(1) Quadratic + quadratic	3.75°	3.34°
(1b) Quadratic + quadratic (modified)	2.97°	2.91°
(2) Charbonnier + Charbonnier	2.69°	2.91°
(2b) Charbonnier + Charbonnier (modified)	2.21°	2.38°
(3) Charbonnier + Lorentzian	2.76°	2.72°
(3b) Charbonnier + Lorentzian (modified)	2.33°	2.32°
(4) 3×3 FoE (separate $\mathbf{u}$ & $\mathbf{v}$) + Lorentzian	2.04°	2.31°
(5) 5×5 FoE (separate $\mathbf{u}$ & $\mathbf{v}$) + Lorentzian	2.08°	2.41°
(6) $3 \times 3 \times 2$ FoE (joint $\mathbf{u}$ & $\mathbf{v}$) + Lorentzian	1.98°	2.12°

Table 5.1: **Results on synthetic test data set of 36 image sequences generated using our optical flow database:** Average angular error (AAE) for best parameters.

The gradient of the spatial term is given by Eq. (3.24); the gradient for the data term is equivalent to the data term discretization from [Bruhn et al., 2005]. As explained in more detail in Appendix B.1, we can rewrite the gradient of the data term using $\nabla_{\mathbf{w}} E_{\mathrm{D}}(\mathbf{w}) = \mathbf{A}_{\mathrm{D}}(\mathbf{w})\mathbf{w} + \mathbf{b}_{\mathrm{D}}(\mathbf{w})$, where $\mathbf{A}_{\mathrm{D}}(\mathbf{w})$ is a large, sparse matrix and $\mathbf{b}_{\mathrm{D}}(\mathbf{w})$ is a vector, both of which depend on $\mathbf{w}$. Similarly, we can rewrite the gradient of the spatial term as $\nabla_{\mathbf{w}} E_{\mathrm{FoE}}(\mathbf{w}) = \mathbf{A}_{\mathrm{FoE}}(\mathbf{w})\mathbf{w}$, where the matrix $\mathbf{A}_{\mathrm{FoE}}(\mathbf{w})$ is sparse and depends on $\mathbf{w}$. This is also shown in more detail in Appendix B.1. The gradient constraint in Eq. (5.11) can thus be rewritten as

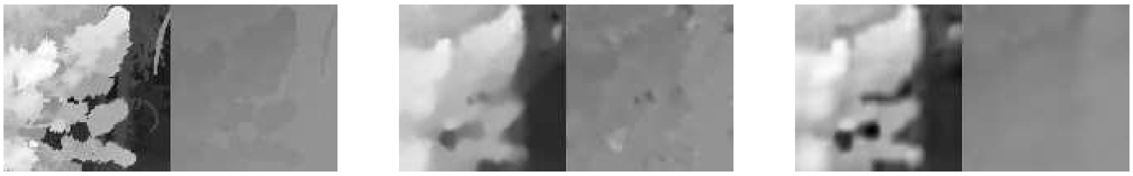
$$[\mathbf{A}_{\mathrm{D}}(\mathbf{w}) + \lambda \cdot \mathbf{A}_{\mathrm{FoE}}(\mathbf{w})]\,\mathbf{w} = -\mathbf{b}_{\mathrm{D}}(\mathbf{w}). \tag{5.12}$$

In order to solve for $\mathbf{w}$, we make Eq. (5.12) linear by keeping $\mathbf{A}_{\mathrm{D}}(\mathbf{w}) + \lambda \cdot \mathbf{A}_{\mathrm{FoE}}(\mathbf{w})$ and $\mathbf{b}_{\mathrm{D}}(\mathbf{w})$ fixed, and solve the resulting linear equation system using a standard technique [Davis, 2004]. After finding a new estimate for $\mathbf{w}$, we linearize Eq. (5.12) around the new estimate and repeat this linearization procedure until a fixed point of the nonlinear equation system is reached.

For all the experiments conducted here, we select the smoothness weight λ by hand as described in more detail alongside each experiment. Recently, Krajsek and Mester [2006] proposed a technique for automatically selecting this parameter for each input sequence, however did so for simpler models of spatial smoothness. It seems worthwhile to study whether their technique can be extended to high-order flow priors as introduced here.

5.4.1 Experimental evaluation

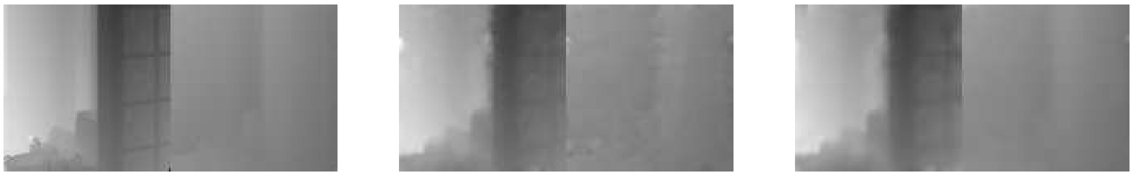
To evaluate the proposed method, we performed a series of experiments with both synthetic and real data. The quantitative evaluation of optical flow techniques suffers from the problem that only a few image sequences with ground truth optical flow data are available.



(a) Ground truth flow.

(b) Standard 2D-CLG ($AAE = 10.86°$).

(c) 2D-CLG with FoE spatial term ($AAE = 8.92°$).



(d) Ground truth flow.

(e) Standard 2D-CLG ($AAE = 1.57°$).

(f) 2D-CLG with FoE spatial term ($AAE = 1.46°$).

(g) Ground truth flow.

(h) Standard 2D-CLG ($AAE = 1.59°$).

(i) 2D-CLG with FoE spatial term ($AAE = 1.83°$).

Figure 5.10: **Optical flow estimation for 3 representative test scenes from 36 used for evaluation.** The horizontal motion is shown on the left side of each flow field; the vertical motion is displayed on the right. The middle column shows the flow estimation results from algorithm *(2b)*, the right column shows the flow estimation results from algorithm *(4)*. The bottom row shows an example where algorithm *(4)* does not perform as well as algorithm *(2b)*.

The first part of our evaluation thus relies on synthetic test data. To provide realistic image texture we randomly sampled 36 (intensity) images from a database of natural images [Martin et al., 2001] and cropped them to 140×140 pixels. The images were warped with randomly sampled, synthesized flow from a separate test set, and the final image pair was cropped to 100×100 pixels. The flow fields for testing were generated from range data and camera motion using the same algorithm as used to generate the training data; the details of this procedure are described in Section 5.2.1. While not identical, the testing data is thus built on the same set of assumptions such as rigid scenes and no independent motion.

We ran 6 different algorithms on all the test image pairs: *(1)* The 2D-CLG approach with quadratic data and spatial terms; *(2)* The 2D-CLG approach with Charbonnier data and spatial terms as used in [Bruhn et al., 2005]; *(3)* The 2D-CLG approach with Lorentzian data term and Charbonnier spatial term; algorithms *(4-6)* are all based on the 2D-CLG approach with Lorentzian data term and an FoE spatial term. *(4)* uses a 3×3 FoE model with separate models for $\mathbf{u}$ and $\mathbf{v}$; *(5)* uses a 5×5 FoE model again with separate models for $\mathbf{u}$ and $\mathbf{v}$; *(6)* uses a $3 \times 3 \times 2$ FoE model that jointly models $\mathbf{u}$ and $\mathbf{v}$. Two different variants of algorithms *(1-3)* were used; one using the discretization described in [Bruhn et al., 2005], and the other using the (correct) discretization actually used by the authors in their experiments [Bruhn, 2006]. Details are provided in Appendix B.2. The corrected versions are marked as *(1b)*, *(2b)*, and *(3b)* respectively.

The Charbonnier robust error function has the form $\rho(x) = 2\beta^2\sqrt{1 + x^2/\beta^2}$, where β is a scale parameter. The Lorentzian robust error function is related to the t-distribution and has the form $\rho(x) = \log(1 + \frac{1}{2}(x/\beta)^2)$, where β is its scale parameter. For all experiments in this chapter, we chose a fixed integration scale for the structure tensor ($\sigma = 1$). For methods *(2)* and *(3)* we tried $\beta \in \{0.05, 0.01, 0.005\}$[4] for both the spatial and the data term and report the best results. For methods *(3-6)* we fixed the scale of the Lorentzian for the data term to $\beta = 0.5$. For each method we chose a set of 10 candidate λ values (in a suitable range), which control the relative weight of the spatial term. Each algorithm was run using each of the candidate λ values. Using a simple MATLAB implementation, running method *(1)* on one frame pair takes on average $1.8s$, method *(2)* takes $15.7s$, and method *(4)* averages at $218.9s$. The increase in computational effort from using the FoE model stems from the fact that the linear equation systems corresponding to Eq. (5.12) are less sparse.

We measure the performance of the various methods using the average angular error [Barron et al., 1994]

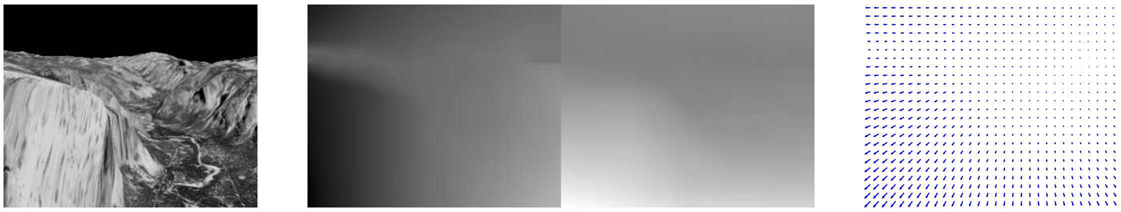
$$AAE(\mathbf{w}_e, \mathbf{w}_t) = \arccos\left(\frac{\mathbf{w}_e^{\mathrm{T}}\mathbf{w}_t}{||\mathbf{w}_e|| \cdot ||\mathbf{w}_t||}\right), \tag{5.13}$$

where $\mathbf{w}_e = (\mathbf{u}_e, \mathbf{v}_e, 1)$ is the estimated flow and $\mathbf{w}_t = (\mathbf{u}_t, \mathbf{v}_t, 1)$ is the true flow. We exclude 5 pixels around the boundaries when computing the AAE. Table 5.1 shows the mean and the standard deviation of the average angular error for the whole test data set (36 flow fields). The error is reported for the λ value that gave the lowest average error on the whole data set, i. e., the parameters are not tuned to each individual test case to emphasize generality of the method. Figure 5.10 shows 3 representative results from this benchmark. We can see that the FoE model recovers smooth gradients in the flow rather well, and that the resulting flow fields look less "noisy" than those recovered by the baseline CLG algorithm. On the other hand, motion boundaries sometimes appear blurred in the results from the FoE model. We presume that this is in part due to local minima in the non-convex energy from Eq. (5.10). Future work should address the problem of inference with such complex non-convex energies as discussed in more detail in Section 5.5.

The quantitative results in Table 5.1 show that the FoE flow prior improves the flow

---

[4] This parameter interval is suggested in [Bruhn et al., 2005].



(a) Frame 8 from image sequence. (b) Estimated optical flow with separate **u** and **v** components. Average angular error 1.43°. (c) Estimated optical flow as vector field.

Figure 5.11: **Optical flow estimation:** Yosemite fly-through.

estimation error on this synthetic test database. A Wilcoxon signed rank test for zero median shows that the difference between the results for *(2b)* and *(4)* is statistically significant at a 95% confidence level ($p = 0.0015$). We furthermore find that an FoE prior with 5×5 cliques does not provide superior performance over a 3×3 model; the performance in fact deteriorates slightly, which may be attributable to local minima in the inference process, but the difference is not statistically significant (using the same test as above). Given that the derivatives of horizontal and vertical flow are largely independent (see Section 5.2.3), it is not surprising that the joint model for horizontal and vertical flow only gives a small performance advantage, which is furthermore not statistically significant. In contrast to methods *(2)* and *(3)* all FoE priors do not require any manual tuning of the parameters of the prior; instead the parameters are learned from training data. Only the λ value requires tuning for all 6 techniques.

Learned pairwise model. We also evaluated the performance of a learned pairwise model of optical flow based on Eq. (5.7) with t-distribution potentials, roughly equivalent to the model of [Black and Anandan, 1996]. We trained the pairwise model using contrastive divergence as described in Section 5.3, and estimated flow as above; aside from the learned prior, the setup was the same as for algorithm *(3b)*. Using the same benchmark set we found that the learned pairwise model lead to worse results than the hand-tuned CLG model (the average AAE rose to 4.28°). Compared to the convex prior employed in *(3b)* the learned prior model is non-convex, which may cause the algorithm to suffer from local optima in the objective. To investigate this further, we initialized both algorithms with the ground truth flow to compare their behavior; both algorithms were run with the best parameters as found based on the regular (non-ground truth) initialization. We found that both models perform almost equally well in this case (0.88° AAE for the learned pairwise model compared to 0.91° for method *(3b)*). This illustrates the need for developing better inference techniques for optical flow estimation with non-convex models.

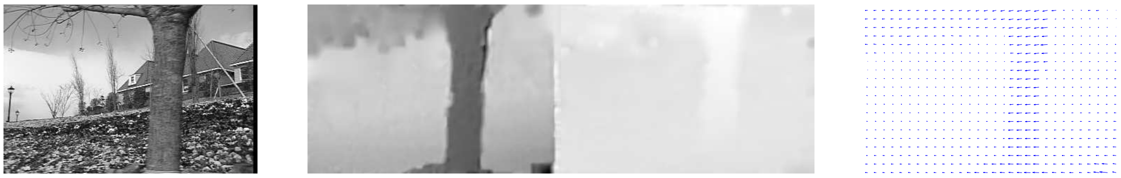
Other sequences. In a second experiment, we learned an FoE flow prior for the Yosemite sequence [Barron et al., 1994], a computer generated image sequence (version without the cloudy sky). First we trained the FoE prior on the ground truth data for the Yosemite sequence, omitting frames 8 and 9 which were used for evaluation. To facilitate comparisons with other methods, we used the image derivative filters given in [Bruhn et al., 2005] for this particular experiment. Estimating the flow with the learned model and the same data term as above gives an average angular error of 1.43° (standard deviation 1.51°). This is 0.19° better than the result reported for the standard two frame CLG method (see [Bruhn et al., 2005]), and 0.15° better than the result of Mémin and Pérez [2002], which is as far as we are aware currently the best result for a two frame method. While training on the remainder of the Yosemite sequence may initially seem unfair, most other reported results for this sequence rely on tuning the method's parameters so that one obtains the best results on a particular frame pair. Figure 5.11 shows the computed flow field. We can see that it seems rather smooth, but given that the specific training data contains only very few discontinuities, this is not very surprising. In fact, changing the λ parameter of the algorithm so that edges start to appear leads to numerically inferior results.

An important question for a learned prior model is how well it generalizes. To evaluate this, we used the model trained on our synthetic flow data to estimate the Yosemite flow. Using the same parameters described above, we found that the accuracy of the flow estimation results decreased to 1.60° average angular error (standard deviation 1.63°). While this result is not as good as that from the Yosemite-specific model, it still slightly outperforms the regular 2D-CLG approach, while not requiring any manual tuning of the prior parameters. This raises two interesting questions: Is the generic training database not fully representative of the kinds of geometries or motions that occur in the Yosemite sequence? Or is it the case that the Yosemite sequence is not representative of the kinds of image motions that occur naturally? This further suggests that future research should be devoted to determining what constitutes generic optical flow and to developing more comprehensive benchmark data sets for flow estimation. In particular, a better data set would include realistic models of the appearance of the scene in addition to the motion. Moreover, this suggests that particular care must be taken when designing a representative optical flow database.

In a final experiment, we evaluated the FoE flow prior on two real image sequences[5]. Figure 5.12 shows the first frame from the "flower garden" sequence as well as the estimated flow. The sequence has two dominant motion layers, a tree in the foreground and a background, with different image velocities. Figure 5.13 shows one frame and flow estimation

---

[5] The "Mobile & Calendar" sequence was kindly provided by Étienne Mémin.

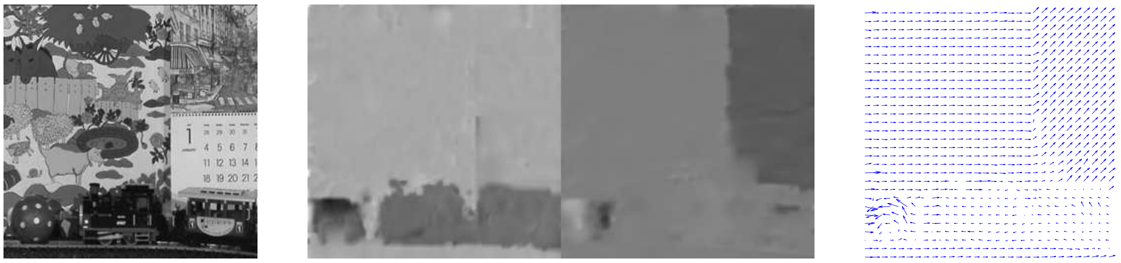


(a) Frame 1 from image sequence.

(b) Estimated optical flow with separate **u** and **v** components.

(c) Estimated flow as vector field.

Figure 5.12: **Optical flow estimation:** Flower garden sequence.



(a) Frame 1 from image sequence.

(b) Estimated optical flow with separate **u** and **v** components.

(c) Estimated flow as vector field.

Figure 5.13: **Optical flow estimation:** Mobile & calendar sequence.

results for the "mobile & calendar" sequence (downsampled to 256 × 256), which exhibits independent motion of the calendar, the train, and the ball in front of the train. In both cases we applied the FoE model as trained on the synthetic flow database and used the parameters as described above for model *(4)*. Both figures show that the obtained flow fields qualitatively capture the motion and object boundaries well, even in case of independent object motion, which was not present in the training set.

5.5 Summary and Future Work

In this chapter we presented a novel database of optical flow as it arises when realistic scenes are captured with a hand-held or car-mounted video camera. This database allowed us to study the spatial and temporal statistics of optical flow. We found that "natural" optical flow is mostly slow, but sometimes fast, as well as mostly smooth, but sometimes discontinuous. Furthermore, the derivative statistics were found to be very heavy-tailed, which likely explains the success of previous flow estimation techniques based on robust potential functions. Moreover, the flow data enabled us to learn prior models of optical flow using the Fields-of-Experts framework. We integrated a learned FoE flow prior into a recent, accurate optical flow algorithm and obtained statistically significant accuracy improvements on a synthetic test set. While our experiments suggest that the training database may not yet be representative of the image motion in certain sequences, we believe that this is an important step towards studying and learning the spatial statistics of optical flow.

There are many opportunities for future work that build on the proposed prior and the database of flow fields. Since many of them are specific to the application of optical flow, we will discuss them here rather than in the context of the full work. For example, Calow et al. [2004] point out that natural flow fields are inhomogeneous; for example, in the case of human motion, the constant presence of a ground plane produces quite different flow statistics in the lower portion of the image than in the upper portion. The work of Torralba [2003] on estimating global scene classes could be used to apply scene and region appropriate flow priors.

It may also be desirable to learn application-specific flow priors (e. g., for automotive applications). This suggests the possibility of learning multiple categories of flow priors [e. g., Torralba and Oliva, 2003] and using these to classify scene motion for applications in video databases.

So far, inference with this optical flow model has relied on iteratively solving linear equation systems, which suffers from the fact that the data and spatial term are non-convex. In case of pairwise MRFs for regularization, other work has relied on annealing techniques to partly overcome this problem [Black and Anandan, 1996]. However, from recent research on stereo problems, it has become apparent that better inference techniques such as graph cuts or belief propagation are particularly good at minimizing the corresponding non-convex energies [Szeliski et al., 2006]. The findings in this chapter suggest the need for non-convex regularization models in optical flow computation. It is clear that this complicates the computational techniques needed to recover optical flow, but the success in stereo problems and some initial successes in optical flow [Barbu and Yuille, 2004] suggest that research into better inference techniques for optical flow should be a fruitful area for future work.

A natural extension of this work is the direct recovery of structure from motion. We can exploit our training set of camera motions to learn a prior over 3D camera motions and combine this with a spatial prior learned from the range imagery. The prior over 3D motions may help regularize the difficult problem of recovering camera motion given the narrow field of view and small motions present in common video sequences.

Future work must also consider the statistics of independent, textural, and nonrigid motion. Here obtaining ground truth is more problematic. Possible solutions involve obtaining realistic synthesized sequences from the film industry or hand-marking regions of independent motion in real image sequences.

Finally, a more detailed analysis of motion boundaries is warranted. In particular, this current flow prior does not explicitly encode information about the occluded/unoccluded surfaces or the regions of the image undergoing deletion/accretion. Future work may also explore the problem of jointly learning motion and occlusion boundaries using energy-based models such as Fields of Experts [cf., Ross and Kaelbling, 2005].

CHAPTER 6

Summary and Outlook

The need for imposing prior knowledge is omnipresent in low-level vision applications due to their frequently underconstrained and ill-posed nature. While there are many ways of imposing prior knowledge, we argued that probabilistic models of prior knowledge are particularly suitable for this task. Bayesian probabilistic models very naturally allow us to treat the uncertainty inherent in many low-level vision problems, let us maintain uncertainty of the solution, enable us to combine information from several sources in a principled way, and allow us access a large literature on efficient algorithms for inferring the output. Probably the most important advantage of probabilistic models for low-level vision from the viewpoint taken here is that they let us exploit the statistical properties of the data, for example through careful choice of the form of the model, or by learning the model parameters from training data. This enables us to make informed decisions on what makes good models, and allows us to avoid tedious hand-tuning of parameters that has been prevalent in non-probabilistic approaches.

As we have seen, there are a wide variety of probabilistic approaches for modeling prior knowledge in low-level vision, including patch-based models, Wavelet-based approaches, as well as Markov random fields, which have been very popular in the literature. MRFs have important advantages over patch-based models or many Wavelet approaches, for example in that they model the entire image, or flow field, depth map, etc. This is important, because it allows the use of MRF priors in a wide variety of situations, such as with missing data, where many other models require complicated workarounds to still be applicable. Also, there is a large literature on efficient learning and inference methods that vision researchers can rely on when employing MRF models. While MRFs can quite easily be applied to a wide range of problems, they have not been without problems. To this date, most MRF models in low-level vision are hand-defined; in particular their parameters are chosen by hand, even though probabilistic models in principle allow us to avoid this. Moreover, the statistical properties of the data are often only vaguely considered when designing these models. Nonetheless, it is possible to motivate MRFs from the statistical properties of the data and to learn the model parameters from training data. This makes them more principled and less tedious to use, and in many cases gives us better application performance in addition.

Despite all that, MRF models in low-level vision have recently fallen short of other approaches in terms of their application performance; for example, Wavelet-based methods outperform traditional MRFs in image denoising. We showed that this is in a very large part due to the fact that typical MRFs in low-level vision are pairwise models, which are based on modeling horizontally and vertically neighboring pairs of pixels. The kinds of spatial structures that can be expressed with such a simple MRF are too limited to allow the model to capture complicated structures that exist in real images, flow fields, and so on. To overcome this limitation, this dissertation introduced Fields of Experts, a high-order MRF suitable for modeling various types of low-level scene representations such as images, optical flow, and others. FoEs model spatial structures using extended, overlapping neighborhoods (maximal cliques), for example of 5×5 pixels. The challenge in modeling spatial data with high-order MRFs is to find suitable potential functions that properly describe the "compatibility" of the pixels in the maximal cliques. We addressed this by combining insights from modeling small, fixed size image patches with high-order MRFs, which allows us to model images (and other scene representations) of an arbitrary size, while at the same time having a model that is much more expressive than standard pairwise MRFs. In particular, we showed how Products of Experts can be used to define the potentials of a high-order MRF. In this formulation each expert is a simple parametric model that describes the response to a linear filter defined on the clique pixels. The parametric form of the resulting FoE not only allows us to learn the parameters of the experts from training data, but also lets us learn a rich set of filters from training data. This distinguishes FoEs from previous high-order MRFs in vision, such as the FRAME model. Learning in Markov random fields, particularly in MRFs of high-order, is computationally very challenging. Despite that we showed how contrastive divergence, an efficient alternative to maximum likelihood estimation, can be used to estimate the parameters of the FoE prior from data (i. e., the expert parameters and the filters). In contrast to FRAME, which adds filters to the model in a greedy one-by-one fashion (taken from a hand-defined set), FoEs allow all filters to be learned simultaneously. While this procedure is still computationally expensive, it is fast enough to be practical. Another advantage of the parametric form of FoEs is that it allows for efficient approximate inference algorithms based on continuous local optimization.

After introducing the model in a general form, we showed how it can be applied to two different kinds of low-level scene representations, natural images and optical flow. We showed how FoEs address some of the important modeling challenges that occur in natural images, particularly complex long-range dependencies. We motivated the use of Student t-experts from the statistics of natural images, and showed how various FoE models can be trained on large databases of natural images. One very interesting observation made from these models is that the filters that were learned are of high spatial frequency and are seemingly less structured than other filters that have commonly been used in image processing applications. Through study of image restoration applications we showed that these unusual filters are in fact important to obtaining good application performance. Furthermore, we showed that learning the filters alongside the other parameters of the model leads to a significant advantage in application performance. Our results reemphasize findings in previous work [e. g., Black et al., 1998] that non-convex regularization is important for achieving very good application performance. In particular, we showed that a model based on convex (Charbonnier) experts performed notably worse than a non-convex model based on Student t-experts. We demonstrated the use of the FoE model of natural images in two image restoration applications: image denoising and image inpainting. In both cases, we showed that FoEs perform at or very close to the state-of-the-art in either application, while being more generic and more widely applicable than the current alternatives, such as Wavelet methods in image denoising. In either application, Fields of Experts substantially outperformed pairwise Markov random field models, which clearly demonstrates the increased modeling power of high-order MRFs. It is particularly noteworthy that such levels of application performance had previously not been demonstrated with generic MRF priors. We believe that this is a very important step in making Markov random field priors practical for wide deployment in a variety of low-level vision applications.

To further reinforce this assessment, we also studied the application of Fields of Experts to modeling optical flow. Optical flow poses another challenge in that ground truth training data is not as easily available as is the case for natural images. To overcome this problem, we showed how realistic optical flow data can be synthesized from realistic 3D scene models and databases of camera motion. While the statistics of other low-level scene representations such as images and depth maps had been studied before, this database allowed us to perform the first comprehensive analysis of the statistical properties of optical flow. The analysis showed some important properties of optical flow: For example, the particular shape of the empirical derivative histograms of optical flow provides an explanation for the success of non-convex regularization techniques in optical flow estimation. Beyond this analysis, the database also enabled us to train an FoE model of optical flow, which in contrast to previous work is based on extended pixel neighborhoods. We integrated this spatial prior of optical flow into a recent, competitive optical flow algorithm, and showed that the richer modeling capabilities of FoEs lead to noticeable performance improvements in optical flow estimation when compared to previous regularization approaches that are closely related to pairwise MRFs. This once again demonstrates the increased modeling power of high-order MRFs over standard pairwise models. The success in modeling different low-level scene representations such as images and optical flow suggests that FoEs not only apply to other scene representations as well, but also that we can expect significant performance improvements from doing so.

6.1 Limitations and Outlook

In this dissertation we saw that high-order MRF models, and in particular Fields of Experts, have a greatly increased modeling power over traditional pairwise MRFs. Moreover, FoEs exhibit application performance levels that are competitive with the state-of-the-art in a variety of applications, while being very generic and widely applicable. Nevertheless, the presented approach has a number of limitations, which motivate future research in this area.

One shortcoming of the FoE model is that it does not properly reproduce all of the statistical properties of natural images that have been considered important in the literature. In particular, we showed that while the marginal statistics of model samples are somewhat heavy-tailed, they are not nearly as heavy-tailed as the empirical marginals of the training data. As we discussed, this may be attributable to the parametric form of the experts that we chose. Future work should thus consider developing other parametric forms or develop other modifications of the model in order to overcome this limitation. Such possibilities include the use of Gaussian scale mixtures [Wainwright and Simoncelli, 2000; Weiss and Freeman, 2007b] or spline-based representations as expert densities. One interesting aspect of this limitation is that pairwise MRF models actually represent marginal derivative statistics well (see Section 4.2), yet, as we showed, perform substantially worse in several low-level vision applications. This suggests that further research on the statistics of natural images is necessary in order to determine which of the statistical properties of natural images are most important for their modeling. The results in this dissertation suggest that properly accounting for complex spatial structures may be more important than properly modeling marginal statistics.

Another limitation of the FoE model with regards to natural image statistics is that it is not scale invariant. As we discussed, natural images have statistics that are mostly invariant to changes in the spatial scale. FoEs cannot model this property, because the filters only operate at a single spatial scale and are limited in size by computational considerations. In order to extend the model toward improved scale invariance, future research should consider both larger filters, and more importantly also hierarchical multiscale models. FoEs with larger filters will require more efficient techniques for learning and inference. Hierarchical models, where several spatial scales are modeled simultaneously, have already been considered in different contexts [e. g., Luettgen et al., 1993; Zhu et al., 1998; Willsky, 2002; Sallee and Olshausen, 2003; Ferreira et al., 2005], but so far either their application performance or their applicability due to very high computational effort has often been limited. Our hope is that future work on efficient learning and inference algorithms will enable the development of scale-invariant extensions of Fields of Experts.

Beyond this, further studies of efficient learning and inference techniques would also benefit the model as proposed. Here, learning relied on contrastive divergence and a fairly generic sampling procedure within contrastive divergence, which has enabled us to train models with filter sizes of up to 7×7. Either applying known efficient sampling procedures, such as the Swendsen-Wang method [Barbu and Zhu, 2005], or devising domain specific efficient sampling algorithms could help shorten the training time for the models we studied, or allow for FoEs with larger filters. It would also be worth investigating alternative learning techniques, including ones that avoid sampling. Score matching [Hyvaärinen, 2005], for example, is a recently proposed method for training non-normalized product models, such as a Product of Experts. FoEs could also be trained with this technique, but some of our initial experiments suggested that FoEs trained with score matching have inferior application performance when compared to models trained using contrastive divergence. Nevertheless, this issue warrants further study.

As we discussed, non-convex expert functions in principle lead to superior results, but make probabilistic inference with FoE models difficult. In our applications we thus relied on simple, local optimization methods, but some of our initial experiments indicated that these may not be appropriate for all categories of interesting low-level vision applications. For example in image super-resolution, gradient ascent methods often do not lead to satisfactory results, even in the pairwise MRF case, where belief propagation, on the other hand, was shown to be successful [Tappen et al., 2003]. Because of that, it seems very worthwhile to study more efficient probabilistic inference techniques, such as belief propagation or graph cut methods in the context of high-order MRF models. Recent progress has already been made on belief propagation in the case of relatively small 2×2 cliques [Lan et al., 2006; Potetz, 2006], but larger clique sizes are clearly desirable.

Another limiting aspect of the FoE model is that it is homogeneous; that is the clique potentials are the same throughout the entire image (or other dense scene representation). While this has the important advantage that it keeps the number of model parameters manageable, as they are shared across all cliques, inhomogeneous variants may nevertheless be useful in various situations. For example in photographic images, the top part of an image often depicts the sky, which has different statistical properties from the objects in the bottom part of the image. Moreover, if we had a higher-level model of scene understanding that, for example, captures certain scene primitives, we may want to couple this high-level description with a random field model of the image. To do that, the potentials of the random field could, for example, depend on the presence or absence of scene primitives at a particular location. For both of these cases, it would be helpful to develop extensions of the FoE that have additional latent variables that control the behavior of each clique. In case of spatially varying structures, these latent variables would have a spatially dependent prior associated with them and may be marginalized out. In case of coupling a low-level model with high-level primitives, the latent variables would be coupled to the representation of the primitives. It has already been shown in other contexts that imposing additional structure on latent variables is both feasible and desirable from a performance point of view [Welling et al., 2003; Lyu and Simoncelli, 2007]. In case of the FoE, structure on the latent variables may be imposed through an interpretation of the expert densities as Gaussian scale mixtures [cf. Welling et al., 2003].

One problem that we encountered in the evaluation of the FoE model is that the likelihood of training and test data sets cannot directly be computed, because the normalization constant cannot be computed in closed form. This is problematic, because we cannot directly evaluate the quality of FoEs, but instead have to measure it indirectly by applying the model to some low-level vision application and measuring the performance there. If all we care about is the performance on a particular application, then this is not a limitation, but from a probabilistic modeling point of view, this is undesirable. It would thus be very interesting to develop bounds on the partition function, which in turn would allow putting bounds on the training or test likelihood. This would allow a more rigorous evaluation of the quality of FoE models. Such an evaluation is not only interesting for comparing FoEs to other models without having to resort to indirect measurements such as application performance, but also because it would enable us to compare FoEs with different parameters. Since we could, for example, compare models with different filters, this would help with understanding the important properties of the model. Weiss and Freeman [2007b] derived bounds on the partition function based on modeling the experts as Gaussian scale mixtures. This allowed them to show that the unusual nature of the FoE filters is not accidental, but actually leads to higher likelihoods than some more common and more regular filters. This bound already provides a major contribution toward a better understanding of FoE models, and further research on tighter bounds would only improve this.

Beyond improving and understanding FoEs in a general sense, it would also be very interesting to apply Fields of Experts to other vision applications and even to other domains. By making MRFs much richer, many low-level vision problems can be revisited with an expectation of improved results. The developed FoE model of natural images is, for example, applicable to image super-resolution, image sharpening, and graphics applications such as image based rendering and others. But there are also interesting future applications of FoE models beyond natural images and optical flow. One obvious further application to explore is that of scene depth. Modeling scene depth is interesting for two main applications, stereoscopic reconstruction [Marr and Poggio, 1976] and structure-from-motion recovery [Hanna, 1991]. When formulated in a Bayesian paradigm similar to the models in the previous chapters, both applications require prior knowledge about scene depth. Many state-of-the-art stereo techniques rely on pairwise Markov random fields, some with a simple 4-connected grid and others with highly connected, but pairwise neighborhoods [Scharstein and Szeliski, 2001]. The study of range images has shown that scene depth shares many basic statistical properties with intensity images [Huang et al., 2000], which suggests that Fields of Experts are well suited for modeling scene depth. The developed framework may also be applicable to surface reconstruction problems in vision and graphics. Diebel et al. [2006], for example, learn a simple Bayesian prior over neighboring surface normals on a 3D mesh. It would be quite interesting to extend the FoE model developed here to allow for high-order cliques on such triangular meshes.

Outside of computer vision, there are a number of other types of data that have properties similar to images. High-order MRF models such as the FoE may, for example, enable better processing of audio signals, or similar types of temporal data. Spatial data including geographic measurements, or measurements of material properties may also benefit from the richer modeling capabilities of high-order MRFs.

6.2 Final Thoughts

This dissertation has shown that high-order Markov random fields are very powerful models for capturing prior knowledge in low-level vision applications. In particular, we demonstrated that they are significantly more powerful than pairwise Markov random fields, because of their ability to model richer spatial structures over large neighborhoods of pixels. While both learning and inference in high-order MRFs are substantially more difficult than in pairwise models, we showed that high-order MRFs can be made practical and lead to performance improvements in real-world applications. At least in part inspired by the power and versatility of the FoE model, a number of other researchers have recently started working on various aspects of high-order MRFs [Potetz, 2006; Rother, 2007; Weiss and Freeman, 2007a; Torr, 2007].

The Fields-of-Experts model developed in this dissertation is only the first step toward making wider use of high-order models in computer vision. On one hand, a number of limitations of the approach should be overcome in the future. On the other hand, complex statistical dependencies do not just exist in low-level vision problems, but also in higher-level models of scene understanding, for example between various parts of an object. While the techniques developed here may not directly apply to these very different kinds of scene representations, we nevertheless think that the advantages of modeling with high-order cliques will become important in high-level scene understanding as well.

All in all, this is a very exciting time to be working in this area.

APPENDIX A

Notation

Scalars	$a, b, c, \dots$
Vectors	$\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots$
Matrices	$\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$
Elements of vectors	$x_1, y_i, z_{2j+1}, \dots$
Column vectors of matrices	$\mathbf{A}_1, \mathbf{B}_i, \dots$
Transposed vector and matrix	$\mathbf{x}^{\mathrm{T}}, \mathbf{A}^{\mathrm{T}}$
Scalar-valued functions	$f(x), g(\mathbf{y}), h(\mathbf{A})$
Vector-valued functions	$\mathbf{f}(x), \mathbf{g}(\mathbf{x}), \mathbf{h}(\mathbf{A})$
First and second derivative of (scalar) function $f(x)$	$\frac{df(x)}{dx},\ \frac{d}{dx}f(x),\ \frac{d^2 f(x)}{dx^2}$
Abbreviated version (also elementwise)	$f'(x),\ f''(x),\ f'(\mathbf{x}),$
Partial derivative of (scalar) function $f(\mathbf{x})$	$\frac{\partial f(\mathbf{x})}{\partial x_i},\ \frac{\partial}{\partial x_i}f(\mathbf{x}),\ \partial_{x_i} f(\mathbf{x})$
Gradient of a scalar function $f(\mathbf{x})$ w. r. t. $\mathbf{x}$	$\nabla_{\mathbf{x}} f(\mathbf{x}),\ \nabla f(\mathbf{x})$
Convolution of image $\mathbf{x}$ with mask $\mathbf{G}$	$\mathbf{G} * \mathbf{x}$
Probability mass function (discrete)	$P(x)$
Probability density function (continuous)	$p(x)$
Conditional probability (density) of x given y	$P(x \vert y),\ p(x \vert y)$
Probability (density) of x given parameters θ	$P(x;\, \theta),\ p(x;\, \theta)$
Expectation value of $f(x)$ w.r.t. probability (density) $p(x)$	$E_{p(x)}[f(x)],\ E_p[f(x)],$ $\langle f(x) \rangle_{p(x)},\ \langle f(x) \rangle_p$
Kullback-Leibler (KL) divergence between p and q	$D(p(x) \vert\vert q(x))$
Mutual information between x and y	$I(x; y)$

Table A.1: **Commonly used mathematical notation.**

APPENDIX B

Technical Details of Optical Flow Estimation

B.1 Derivation Details

This section describes in more detail how the nonlinear equation system for flow estimation in Eq. (5.12) can be derived.

B.1.1 Data term

Starting from the continuous data term of the CLG approach [Bruhn et al., 2005]

$$\int_I \rho_\mathrm{D}\left(\sqrt{\mathbf{w}^\mathrm{T}\mathbf{K}_\sigma(I)\mathbf{w}}\right)\,\mathrm{d}x\,\mathrm{d}y,$$

we first obtain a simple discretization of the data term energy

$$E_\mathrm{D}(\mathbf{w}) = \sum_{i=1}^{K} \rho_\mathrm{D}\left(\sqrt{(u_i, v_i, 1)\cdot\mathbf{K}_i\cdot(u_i, v_i, 1)^\mathrm{T}}\right). \tag{B.1}$$

Here, $\mathbf{w} = (u_1, \dots, u_K, v_1, \dots, v_K)^\mathrm{T}$ is the stacked vector of all horizontal velocities $\mathbf{u}$ and all vertical velocities $\mathbf{v}$; $\mathbf{K}_i$ is the structure tensor $\mathbf{K}_\sigma(I)$ evaluated at pixel i. The structure tensor $\mathbf{K}_i$ is a 3×3 tensor with entries K_{kli}. We can now take partial derivatives with respect to u_i and v_i:

$$\begin{aligned}
\frac{\partial}{\partial u_i}E_\mathrm{D}(\mathbf{w}) &= \tilde{\rho}_\mathrm{D}\left(\sqrt{(u_i, v_i, 1)\cdot\mathbf{K}_i\cdot(u_i, v_i, 1)^\mathrm{T}}\right)(K_{11i}u_i + K_{12i}v_i + K_{13i}) && \text{(B.2)}\\
\frac{\partial}{\partial v_i}E_\mathrm{D}(\mathbf{w}) &= \tilde{\rho}_\mathrm{D}\left(\sqrt{(u_i, v_i, 1)\cdot\mathbf{K}_i\cdot(u_i, v_i, 1)^\mathrm{T}}\right)(K_{21i}u_i + K_{22i}v_i + K_{23i}), && \text{(B.3)}
\end{aligned}$$

where we define $\tilde{\rho}_{\mathrm{D}}(y) = \rho'_{\mathrm{D}}(y)/y$. Note that this formulation is equivalent to the data term parts of Eqs. (38) and (39) in [Bruhn et al., 2005].

Finally, we set the partial derivatives to 0 and regroup the terms in matrix-vector form as

$$0 = \nabla_{\mathbf{w}} E_{\mathrm{D}}(\mathbf{w}) = \mathbf{A}_{\mathrm{D}}(\mathbf{w})\mathbf{w} + \mathbf{b}_{\mathrm{D}}(\mathbf{w}). \tag{B.4}$$

In this notation $\mathbf{A}_{\mathrm{D}}(\mathbf{w})$ is a large, sparse matrix that depends on $\mathbf{w}$, and the matrix-vector product $\mathbf{A}_{\mathrm{D}}(\mathbf{w})\mathbf{w}$ is used to express all terms of Eqs. (B.2) and (B.3) that contain K_{11i}, K_{12i}, K_{21i}, and K_{22i}. $\mathbf{b}_{\mathrm{D}}(\mathbf{w})$ is a vector that also depends on $\mathbf{w}$ and contains all terms with K_{13i} and K_{23i}.

B.1.2 Spatial term

From Eq. (3.24), we know that we can write the gradient of the energy of the FoE spatial term with respect to the flow field component as follows:

$$\nabla_{\mathbf{x}} E_{\mathrm{FoE}}(\mathbf{x}) = -\sum_{i=1}^{N} \mathbf{J}_{-}^{(i)} * \boldsymbol{\psi}'(\mathbf{J}^{(i)} * \mathbf{x};\ \boldsymbol{\alpha}_i).$$

Because convolution is a linear operation, we can express the convolution $\mathbf{J}^{(i)} * \mathbf{x}$ as a matrix-vector product of a filter matrix $\mathbf{F}_i$ with the vectorized flow field component $\mathbf{x}$. Similarly, convolution with the mirrored filter $\mathbf{J}_{-}^{(i)}$ can be expressed as a matrix-vector product; we call the corresponding filter matrix $\mathbf{G}_i$. If we remind ourselves that $\boldsymbol{\psi}'(\mathbf{y})$ is the element-wise application of the nonlinearity ψ' to the elements of $\mathbf{y}$, we can rewrite the preceding equation as

$$\nabla_{\mathbf{x}} E_{\mathrm{FoE}}(\mathbf{x}) = -\sum_{i=1}^{N} \mathbf{G}_i \cdot \boldsymbol{\psi}'(\mathbf{F}_i \cdot \mathbf{x};\ \boldsymbol{\alpha}_i). \tag{B.5}$$

Furthermore, we can exploit the form of the nonlinearity and express it as $\psi'(y;\ \boldsymbol{\alpha}_i) = \zeta_i(y) \cdot y$. If we assume that $\boldsymbol{\zeta}_i$ is applied to vectors in an element-wise fashion, we can express the nonlinearity for vectors as follows:

$$\boldsymbol{\psi}'(\mathbf{y};\ \boldsymbol{\alpha}_i) = \mathrm{diag}\{\boldsymbol{\zeta}_i(\mathbf{y})\} \cdot \mathbf{y}, \tag{B.6}$$

where $\mathrm{diag}\{\mathbf{z}\}$ denotes a diagonal matrix with the entries of vector $\mathbf{z}$ on its diagonal. When combining this with the previous step, we obtain that the gradient of the FoE spatial term can be written as

$$\nabla_{\mathbf{x}} E_{\mathrm{FoE}}(\mathbf{x}) = -\sum_{i=1}^{N} \mathbf{G}_i \cdot \mathrm{diag}\{\boldsymbol{\zeta}_i(\mathbf{F}_i \cdot \mathbf{x})\} \cdot \mathbf{F}_i \cdot \mathbf{x} = \left[-\sum_{i=1}^{N} \mathbf{G}_i \cdot \mathrm{diag}\{\boldsymbol{\zeta}_i(\mathbf{F}_i \cdot \mathbf{x})\} \cdot \mathbf{F}_i\right] \mathbf{x}. \tag{B.7}$$

Note that the term in brackets is a large matrix that depends on $\mathbf{x}$, which we denote as $\mathbf{A}_{\text{FoE}}(\mathbf{x})$. It thus follows that $\nabla_{\mathbf{x}} E_{\text{FoE}}(\mathbf{x}) = \mathbf{A}_{\text{FoE}}(\mathbf{x})\mathbf{x}$.

B.2 Incremental Flow Estimation using Pyramids

Global techniques for optical flow such as the ones presented here often use multi-resolution methods based on image pyramids to overcome the limitations of the local linearization of the optical flow constraint, and to estimate flows with large displacements. One issue that arises when employing incremental multi-resolution schemes for optical flow estimation is how to properly take into account the flow estimate from coarser scales. Usually the input sequence is pre-warped with the flow as estimated at a coarser scale. This result is then incrementally refined at a finer scale [Black and Anandan, 1996]. The data term only considers the incremental flow and is thus easy to deal with; the spatial term on the other hand has to consider the combination of the incremental flow and the estimation from the coarser scales, since the spatial term is a prior model of the combined flow and not of the incremental flow.

To that end, we combine the flow estimate $\mathbf{w}$ from the next coarser scale with the incremental flow $\Delta\mathbf{w}$ and obtain the energy

$$E(\Delta\mathbf{w}) = E_{\text{D}}(\Delta\mathbf{w}) + \lambda \cdot E_{\text{FoE}}(\mathbf{w} + \Delta\mathbf{w}) \tag{B.8}$$

that is to be minimized with respect to $\Delta\mathbf{w}$. As in Section 5.4, we set the gradient to zero and rewrite the gradient terms as $\nabla_{\mathbf{w}} E_{\text{D}}(\Delta\mathbf{w}) = \mathbf{A}_{\text{D}}(\Delta\mathbf{w})\Delta\mathbf{w} + \mathbf{b}_{\text{D}}(\Delta\mathbf{w})$ and $\nabla_{\mathbf{w}} E_{\text{FoE}}(\mathbf{w} + \Delta\mathbf{w}) = \mathbf{A}_{\text{FoE}}(\mathbf{w} + \Delta\mathbf{w}) \cdot (\mathbf{w} + \Delta\mathbf{w})$. The resulting equation system can be written as

$$\left[\mathbf{A}_{\text{D}}(\Delta\mathbf{w}) + \lambda \cdot \mathbf{A}_{\text{FoE}}(\mathbf{w} + \Delta\mathbf{w})\right] \Delta\mathbf{w} = -\mathbf{b}_{\text{D}}(\Delta\mathbf{w}) - \lambda \cdot \mathbf{A}_{\text{FoE}}(\mathbf{w} + \Delta\mathbf{w})\mathbf{w}, \tag{B.9}$$

which we solve iteratively as before.

The 2D-CLG discretization given in [Bruhn et al., 2005] (Eqs. (42) and (43)) is missing a term corresponding to $-\lambda \cdot \mathbf{A}_{\text{FoE}}(\mathbf{w} + \Delta\mathbf{w})\mathbf{w}$ (on the RHS of Eq. (B.9)). This is problematic, because without this term the regularization is not properly applied to the combined flow. The original implementor of [Bruhn et al., 2005] has confirmed that this is simply an oversight in the manuscript [Bruhn, 2006], and that the original implementation is in fact based on the correct discretization. Our implementation of the algorithm confirms this in case of the Yosemite sequence. In the notation of [Bruhn et al., 2005], Eq. (42) can be corrected as follows:

$$0 = \sum_{j \in \mathcal{N}(i)} \frac{\psi_{2i}'^{m} + \psi_{2j}'^{m}}{2} \frac{u_j^m + \delta u_j^m - u_i^m - \delta u_i^m}{h^2} - \frac{\psi_{1i}'^{m}}{\alpha} (J_{11i}^m \delta u_i^m + J_{12i}^m \delta v_i^m + J_{13i}^m); \quad \text{(B.10)}$$

Eq. (43) can be adapted accordingly. For completeness, Section 5.4.1 gives results for both discretizations.

BIBLIOGRAPHY

2d3 Ltd. boujou. `http://www.2d3.com`, 2002.

Ashraf M. Abdelbara and Sandra M. Hedetniemib. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102(1):21–38, June 1998. doi:10.1016/S0004-3702(98)00043-5.

K. Abend, T. J. Harley, and L. N. Kanal. Classification of binary random patterns. *IEEE Transactions on Information Theory*, 11(4):538–544, October 1965.

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover, New York, New York, 1964.

Luis Alvarez, Joachim Weickert, and Javier Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1): 41–56, August 2000. doi:10.1023/A:1008170101536.

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 36(1):99–102, 1974.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, January 2003. doi:10.1023/A:1020281327116.

Suyash P. Awate and Ross T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):364–376, March 2006. doi:10.1109/TPAMI.2006.64.

Adrian Barbu and Alan Yuille. Motion estimation by Swendsen-Wang cuts. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 754–761, Washington, DC, June 2004. doi:10.1109/CVPR.2004.1315107.

Adrian Barbu and Song-Chun Zhu. Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1239–1253, August 2005. doi:10.1109/TPAMI.2005.161.

O. Barndorff-Nielsen, J. Kent, and M. Sørensen. Normal variance-mean mixtures and z distributions. *International Statistical Review*, 50(2):145–159, August 1982.

J[ohn] L. Barron, D[avid] J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994. doi:10.1007/BF01420984.

Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, November 1995.

Anthony J. Bell and Terrence J. Sejnowski. Edges are the 'independent components' of natural scenes. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 831–837, 1997.

Rami Ben-Ari and Nir Sochen. A general framework and new alignment criterion for dense optical flow. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 529–536, New York, New York, June 2006. doi:10.1109/CVPR.2006.25.

H.-J. Bender, R. Männer, C. Poliwoda, S. Roth, and M. Walz. Reconstruction of 3D catheter paths from 2D x-ray projections. In C. Taylor and A. C. F. Colchester, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI'99*, volume 1679 of *Lecture Notes in Computer Science*, pages 981–989. Springer, 1999. doi:10.1007/10704282_107.

Marcelo Bertalmío. Personal communication, 2006.

Marcelo Bertalmío, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Computer Graphics (Proceedings of ACM SIGGRAPH)*, pages 417–424, New Orleans, Louisiana, July 2000. doi:10.1145/344779.344972.

M[arcelo] Bertalmío, L[uminita] Vese, G[uillermo] Sapiro, and S[tanley] Osher. Simultaneous structure and texture image inpainting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 707–712, Madison, Wisconsin, June 2003. doi:10.1109/CVPR.2003.1211536.

Julian Besag. Spatial interaction and the statistical analysis of lattices. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 36(2):192–236, 1974.

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 48(3):259–302, 1986.

Belinda Y. Betsch, Wolfgang Einhäuser, Konrad P. Körding, and Peter König. The world from a cat's perspective - Statistics of natural videos. *Biological Cybernetics*, 90(1):41–50, January 2004. doi:10.1007/s00422-003-0434-6.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, New York, 1995.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Michael J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 296–302, Lahaina, Maui, Hawaii, June 1991. doi:10.1109/CVPR.1991.139705.

Michael J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1): 75–104, January 1996. doi:10.1006/cviu.1996.0006.

Michael J. Black and Anand 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, July 1996. doi:10.1007/BF00131148.

Michael J. Black and Stefan Roth. On the receptive fields of Markov random fields. Cosyne, 2005.

Michael J. Black and Guillermo Sapiro. Edges as outliers: Anisotropic smoothing using local image statistics. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 259–270. Springer, 1999.

Michael J. Black, Guillermo Sapiro, David H. Marimont, and David Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998. doi:10.1109/83.661192.

Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987.

Léon Bottou. Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, number 3176 in Lecture Notes in Artificial Intelligence, pages 146–168. Springer, Berlin, 2004.

Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004. doi:10.1109/TPAMI.2004.60.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11): 1222–1239, November 2001. doi:10.1109/34.969114.

Andrés Bruhn. Personal communication, 2006.

Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, February 2005. doi:10.1023/B:VISI.0000045324.43199.43.

A[ntoni] Buades, B[artomeu] Coll, and J[ean]-M[ichel] Morel. A review of image denoising algorithms, with a new one. *SIAM Multiscale Modeling and Simulation*, 4(2):490–530, 2004. doi:10.1137/040616024.

Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 60–65, San Diego, California, June 2005. doi:10.1109/CVPR.2005.38.

Dirk Calow, Norbert Krüger, Florentin Wörgötter, and Markus Lappe. Statistics of optic flow for self-motion through natural scenes. In U. Ilg, H. Bülthoff, and H. Mallot, editors, *Dynamic Perception*, pages 133–138, 2004.

Miguel Á. Carreira-Perpiñán and Geoffrey E. Hinton. On contrastive divergence learning. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 33–40, Barbados, January 2005.

Francine Catté, Pierre-Louis Lions, Jean-Michel Morel, and Tomeu Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, February 1992. doi:10.1137/0729012.

Rong-Chi Chang, Yun-Long Sie, Su-Mei Chou, and Timothy K. Shih. Photo defect detection for image inpainting. In *Seventh IEEE International Symposium on Multimedia*, pages 403–407, December 2005. doi:10.1109/ISM.2005.91.

Pierre Charbonnier, Laure Blanc-Feéraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 168–172, Austin, Texas, November 1994. doi:10.1109/ICIP.1994.413553.

Pierre Charbonnier, Laure Blanc-Feéraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, February 1997. doi:10.1109/83.551699.

Daniel Cremers and Stefano Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3): 249–265, May 2005. doi:10.1007/s11263-005-4882-4.

Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9): 1200–1212, September 2004. doi:10.1109/TIP.2004.833105.

G. R. Cross and A[nil] K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, January 1983.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, October 1972.

Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, June 2004. doi:10.1145/992200.992205.

Xavier Descombes, Robin D. Morris, Josiane Zerubia, and Marc Berthod. Estimation of Markov random field prior parameters using Markov chain Monte Carlo maximum likelihood. *IEEE Transactions on Image Processing*, 8(7):954–963, July 1999. doi:10.1109/83.772239.

James R. Diebel, Sebastian Thrun, and Michael Brünig. A Bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics*, 25(1):35–59, January 2006. doi:10.1145/1122501.1122504.

David L. Donoho. Denoising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995. doi:10.1109/18.382009.

David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, January 2006. doi:10.1109/TIT.2005.860430.

Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038, Kerkyra, Greece, September 1999. doi:10.1109/ICCV.1999.790383.

Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 895–900, New York, New York, June 2006. doi:10.1109/CVPR.2006.142.

Michael Elad, Boaz Matalon, and Michael Zibulevsky. Image denoising with shrinkage and redundant representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1924–1931, New York, New York, June 2006a. doi:10.1109/CVPR.2006.143.

Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. In *Proc. of EUSIPCO*, Florence, Italy, September 2006b.

Ronan Fablet and Patrick Bouthemy. Non parametric motion recognition using temporal multiscale Gibbs models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 501–508, Kauai, Hawaii, December 2001. doi:10.1109/CVPR.2001.990516.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 261–268, Washington, DC, June 2004. doi:10.1109/CVPR.2004.88.

Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 25(3):787–794, July-August 2006. doi:10.1145/1141911.1141956.

Cornelia Fermüller, David Shulman, and Yiannis Aloimonos. The statistics of optical flow. *Computer Vision and Image Understanding*, 82(1):1–32, April 2001. doi:10.1006/cviu.2000.0900.

Marco Ferreira, David Higdon, Herbert K. Lee, and Mike West. Multi-scale random field models. Working Paper 05-02, Duke University, Institute of Statistics and Decision Sciences, Durham, North Carolina, January 2005.

David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America. Series A, Optics and Image Science*, 4(12):2379–2394, December 1987.

Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1176–1183, Nice, France, October 2003. doi:10.1109/ICCV.2003.1238625.

David J. Fleet, Michael J. Black, Yaseer Yacoob, and Allan D. Jepson. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36 (3):171–193, February 2000. doi:10.1023/A:1008156202475.

David J. Fleet, Michael J. Black, and Oscar Nestares. Bayesian inference of visual motion boundaries. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, pages 139–174. Morgan Kaufmann Pub., 2002.

R[oger] Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition, 1987.

James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 2nd edition, 1990.

William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):24–47, October 2000. doi:10.1023/A:1026501619075.

Brendan J. Frey and Nebojsa Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):1392–1416, September 2005. doi:10.1109/TPAMI.2005.169.

Jerome H. Friedman, Werner Stuetzele, and Anne Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79(387):599–608, September 1984.

Hugo Gävert, Jarmo Hurri, Jaakko Särelä, and Aapo Hyvärinen. FastICA software for MATLAB. `http://www.cis.hut.fi/projects/ica/fastica/`, October 2005. Software version 2.5.

Peter Gehler and Max Welling. Products of "edge-perts". In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 419–426, 2006.

Davi Geiger and Frederico Girosi. Parallel and deterministic algorithms from MRF's: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):401–412, May 1991. doi:10.1109/34.134040.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.

Donald Geman and George Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):367–383, March 1992. doi:10.1109/34.120331.

Donald Geman, Stuart Geman, Christine Graffigne, and Ping Dong. Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):609–628, July 1990. doi:10.1109/34.56204.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, November 1984.

Stuart Geman, Donald E. McClure, and Donald Geman. A nonlinear filter for film restoration and other problems in image processing. *CVGIP: Graphical Models and Image Processing*, 54(2):281–289, July 1992. doi:10.1016/1049-9652(92)90075-9.

Charles J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Proceedings of the 23rd Symposium on the Interface*, Computing Science and Statistics, pages 156–163, Seattle, Washington, April 1991.

Guy Gilboa, Nir Sochen, and Yehoshua Y. Zeevi. Image enhancement and denoising by complex diffusion processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1020–1036, August 2004. doi:10.1109/TPAMI.2004.47.

Georgy L. Gimel'farb. Texture modeling by multiple pairwise pixel interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1110–1114, November 1996. doi:10.1109/34.544081.

Tobias Gisy. Image inpainting based on natural image statistics. Diplom thesis, Eidgenössische Technische Hochschule, Zürich, Switzerland, September 2005.

Ulf Grenander and Anuj Srivastava. Probability models for clutter in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):424–429, April 2001. doi:10.1109/34.917579.

Thomas L. Griffiths and Joshua B. Tenenbaum. Optimal predictions in everyday cognition. *Psychological Science*, 17(9):767–773, September 2006. doi:10.1111/j.1467-9280.2006.01780.x.

Jacques Hadamard. *Lectures on the Cauchy Problem in Linear Partial Differential Equations.* Yale University Press, New Haven, Connecticut, 1923.

K. J. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 156–163, Princeton, New Jersey, October 1991. doi:10.1109/WVM.1991.212812.

Wakako Hashimoto and Koji Kurata. Properties of basis functions generated by shift invariant sparse representations of natural images. *Biological Cybernetics*, 83(2):111–118, July 2000. doi:10.1007/s004220000149.

Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 695–702, Washington, DC, June 2004. doi:10.1109/CVPR.2004.1315232.

Glenn E. Healy and Raghava Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3): 267–276, March 1994. doi:10.1109/34.276126.

Fabrice Heitz and Patrick Bouthemy. Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, December 1993. doi:10.1109/34.250841.

Geoffrey E. Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, Edinburgh, UK, September 1999.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002. doi:10.1162/089976602760128018.

Geoffrey E. Hinton and Yee-Whye Teh. Discovering multiple constraints that are frequently approximately satisfied. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 227–234, Seattle, Washington, August 2001.

Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):381–417, November 1999. doi:10.1214/ss/1009212519.

Thomas Hofmann, Jan Puzicha, and Joachim M. Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):803–818, August 1998. doi:10.1109/34.709593.

Berthold K. P. Horn. *Robot Vision.* MIT Press, 1986.

Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, August 1981. doi:10.1016/0004-3702(81)90024-2.

Jinggang Huang. *Statistics of Natural Images and Models.* PhD thesis, Brown University, 2000.

Jinggang Huang, Ann B. Lee, and David Mumford. Statistics of range images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 324–331, Hilton Head Island, South Carolina, June 2000. doi:10.1109/CVPR.2000.855836.

Aapo Hyvaärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–708, April 2005.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, June 2000. doi:10.1016/S0893-6080(00)00026-5.

Aapo Hyvärinen, Patrik O. Hoyer, and Jarmo Hurri. Extensions of ICA as models of natural images and visual processing. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 963–974, Nara, Japan, April 2003.

Michal Irani. Multi-frame optical flow estimation using subspace constraints. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 626–633, Kerkyra, Greece, September 1999. doi:10.1109/ICCV.1999.791283.

M[ichael] Isard. PAMPAS: Real-valued graphical models for computer vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 613–620, Madison, Wisconsin, June 2003. doi:10.1109/CVPR.2003.1211410.

E. T. Jaynes. *Probability Theory: The Logic of Science.* Cambridge University Press, 2003.

Hailin Jin, A. Yezzi, and Stefano Soatto. Variational multiframe stereo in the presence of specular reflections. In *Proceedings of the First International Symposium on 3D Data Processing, Visualization and Transmission*, pages 626–630, Padova, Italy, June 2002. doi:10.1109/TDPVT.2002.1024128.

Nebojsa Jojic, Brendan J. Frey, and Anitha Kannan. Epitomic analysis of appearance and shape. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 1, pages 34–41, Nice, France, October 2003. doi:10.1109/ICCV.2003.1238311.

I[an] T. Jolliffe. *Principal Component Analysis*. Springer, New York, New York, 2nd edition, 2002.

Michael I. Jordan and Yair Weiss. Graphical models: Probabilistic inference. In M. Arib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, Massachusetts, 2nd edition, 2002.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999. doi:10.1023/A:1007665907178.

Thomas Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, COM-15(1):52–60, February 1967.

R. L. Kashyap and R[ama] Chellappa. Filtering of noisy images using Markov random field models. In *Proceedings of the Nineteenth Allerton Conference on Communication Control and Computing*, pages 850–859, Urbana, Illinois, October 1981.

Charles Kervrann and Jérôme Boulanger. Unsupervised patch-based image regularization and representation. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proceedings of the Ninth European Conference on Computer Vision*, volume 3954 of *Lecture Notes in Computer Science*, pages 555–567. Springer, 2006. doi:10.1007/11744085_43.

Pushmeet Kohli and Philip H.S. Torr. Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proceedings of the Ninth European Conference on Computer Vision*, volume 3952 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 2006. doi:10.1007/11744047_3.

Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the Seventh European Conference on Computer Vision*, volume 2352 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 2002.

Vladmimir Kolmogorov and Carsten Rother. Minimizing non-submodular functions with graph cuts - A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. To appear.

Vladmimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2): 147–159, February 2004. doi:10.1109/TPAMI.2004.1262177.

J[anusz] Konrad and E[ric] Dubois. Multigrid Bayesian estimation of image motion fields using stochastic relaxation. In *Proceedings of the Second IEEE International Conference on Computer Vision*, pages 354–362, Tampa, Florida, December 1988.

Kai Krajsek and Rudolf Mester. On the equivalence of variational and statistical differential motion estimation. In *Southwest Symposium on Image Analysis and Interpretation*, pages 11–15, Denver, Colorado, March 2006. doi:10.1109/SSIAI.2006.1633712.

E. R. Kretzmer. Statistics of television signals. *Bell System Technical Journal*, 31:751–763, July 1952.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loelinger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, February 2001. doi:10.1109/18.910572.

M. Pawan Kumar, P[hilip] H. S. Torr, and A[ndrew] Zisserman. OBJCUT. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 18–25, San Diego, California, June 2005. doi:10.1109/CVPR.2005.249.

M. Pawan Kumar, P[hilip] H. S. Torr, and A[ndrew] Zisserman. Solving Markov random fields using second order cone programming relaxations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1045–1052, New York, New York, June 2006. doi:10.1109/CVPR.2006.283.

Sanjiv Kumar and Martial Hebert. Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–201, June 2006. doi:10.1007/s11263-006-7007-9.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, Williamstown, Massachusetts, June-July 2001.

Xiangyang Lan, Stefan Roth, Daniel P. Huttenlocher, and Michael J. Black. Efficient belief propagation with learned higher-order Markov random fields. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proceedings of the Ninth European Conference on Computer Vision*, volume 3952 of *Lecture Notes in Computer Science*, pages 269–282. Springer, 2006. doi:10.1007/11744047_21.

Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

Yann LeCun and Fu Jie Huang. Loss functions for discriminative training of energy-based models. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 206–213, Barbados, January 2005.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. doi:10.1109/5.726791.

Ann B. Lee and Jinggang Huang. Brown range image database. `http://www.dam.brown.edu/ptg/brid/index.html`, 2000.

Ann B. Lee, David Mumford, and Jinggang 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, January 2001. doi:10.1023/A:1011109015675.

Ann B. Lee, Kim S. Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1-3): 83–103, May 2003. doi:10.1023/A:1023705401078.

Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *Proceedings of the Ninth European Conference on Computer Vision*, volume 3954 of *Lecture Notes in Computer Science*, pages 581–594. Springer, 2006. doi:10.1007/11744085_45.

Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 1, pages 305–312, Nice, France, October 2003. doi:10.1109/ICCV.2003.1238360.

G. D. Lewen, W. Bialek, and R. R. de Ruyter van Steveninck. Neural coding of naturalistic motion stimuli. *Network: Computation in Neural Systems*, 12(3):317–329, March 2001. doi:10.1088/0954-898X/12/3/305.

Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2nd edition, 2001.

Ce Liu, Song Chun Zhu, and Heung-Yeung Shum. Learning inhomogeneous Gibbs model of faces by minimax entropy. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 1, pages 281–287, Vancouver, British Columbia, Canada, July 2001. doi:10.1109/ICCV.2001.10007.

Ce Liu, William T. Freeman, Richard Szeliski, and Sing Bing Kang. Noise estimation from a single image. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 901–908, New York, New York, June 2006. doi:10.1109/CVPR.2006.207.

Hongjing Lu and Alan L. Yuille. Ideal observers for detecting motion: Correspondence noise. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 827–834, 2006.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, April 1981.

Mark R. Luettgen, William C. Karl, Alan S. Willsky, and Robert R. Tenney. Multiscale representations of Markov random fields. *IEEE Transactions on Signal Processing*, 41 (12):3377–3396, December 1993. doi:10.1109/78.258081.

Siwei Lyu and Eero P. Simoncelli. Statistical modeling of images with fields of Gaussian scale mixtures. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19, 2007.

David J. C. MacKay. Failures of the one-step learning algorithm. Published online at `http://www.inference.phy.cam.ac.uk/mackay/abstracts/gbm.html`, 2001.

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 2006. Submitted.

David Marr. *Vision*. W. H. Freeman, 1982.

D[avid] Marr and T[omaso] Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, October 1976.

J. Marroquin, S[anjoy] Mitter, and T[omaso] Poggio. Probabilistic solutions of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82 (397):76–89, March 1987.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 2, pages 416–423, Vancouver, British Columbia, Canada, July 2001. doi:10.1109/ICCV.2001.937655.

Georges Matheron. Modèle séquentiel de partition aléatoire. Technical report, Centre de Morphologie Mathématique, 1968.

Julian J. McAuley, Tibério Caetano, Alex J. Smola, and Matthias O. Franz. Learning high-order MRF priors of color images. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 617–624, Pittsburgh, Pennsylvania, June 2006. doi:10.1145/1143844.1143922.

Talya Meltzer, Chen Yanover, and Yair Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 1, pages 428–435, Beijing, China, October 2005. doi:10.1109/ICCV.2005.110.

Étienne Mémin and Patrick Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, February 2002. doi:10.1023/A:1013539930159.

Thomas Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, January 2001.

Thomas Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, Cambridge, UK, 2005.

Teodor Mihai Moldovan, Stefan Roth, and Michael J. Black. Denoising archival films using a learned Bayesian model. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2641–2644, Atlanta, Georgia, October 2006. doi:10.1109/ICIP.2006.313052.

Teodor Mihai Moldovan, Stefan Roth, and Michael J. Black. Denoising archival films using a learned Bayesian model. Technical Report CS-07-03, Brown University, Department of Computer Science, Providence, Rhode Island, 2007.

John Moussouris. Gibbs and Markov random systems with constraints. *Journal of Statistical Physics*, 10(1):11–33, January 1974. doi:10.1007/BF01011714.

David [B.] Mumford. The Bayesian rationale for energy functions. In B. Romeny, editor, *Geometry-Driven Diffusion in Computer Vision*, pages 141–153. Kluwer Academic, 1994.

David B. Mumford and Jayant M. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–684, July 1989.

David W. Murray and Bernard F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, Ontario, Canada, September 1993.

Radford M. Neal. *Bayesian Learning for Neural Networks.* Number 118 in Lecture Notes in Statistics. Springer, New York, New York, 1996.

Radford M. Neal and Geoffrey E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1998.

Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo E. Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, September 2005. doi:10.1109/TIP.2005.852470.

B[runo] A. Olshausen and D[avid] J. Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7(2):333–339, May 1996. doi:10.1088/0954-898X/7/2/014.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, December 1997. doi:10.1016/S0042-6989(97)00169-7.

Rupert Paget. Strong Markov random field model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):408–413, March 2004. doi:10.1109/TPAMI.2004.1262338.

Rupert Paget and I. Dennis Longstaff. Texture synthesis via a noncausal nonparametric multiscale Markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, June 1998. doi:10.1109/83.679446.

Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, April 2006. doi:10.1007/s11263-005-3960-y.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Francisco, California, 2nd edition, 1988.

Patrick Pérez. Markov random fields and images. *CWI Quarterly*, 11(4):413–437, December 1998.

Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990. doi:10.1109/34.56205.

Lyndsey C. Pickup, Stephen J. Roberts, and Andrew Zisserman. A sampled texture prior for image super-resolution. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, 2004.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1997. doi:10.1109/34.588021.

Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, September 1985. doi:10.1038/317314a0.

Jörg Polzehl and Vladimir G. Spokoiny. Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 62(2):335–354, 2000.

Javier Portilla. Benchmark images. `http://www.io.csic.es/PagsPers/JPortilla/denoise/test_image` 2006a.

Javier Portilla. Image denoising software. `http://www.io.csic.es/PagsPers/JPortilla/denoise/softw` 2006b. Software version 1.0.3.

Javier Portilla and Eero P. Simoncelli. Image denoising via adjustment of wavelet coefficients magnitude correlation. In *Proceedings of the 7th International Conference on Image Processing*, volume 3, pages 277–280, Vancouver, Canada, September 2000. doi:10.1109/ICIP.2000.899349.

Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, November 2003. doi:10.1109/TIP.2003.818640.

Brian Potetz. Personal communication, 2006.

Charles A. Poynton. Rehabilitation of gamma. In B. Rogowitz and T. Pappas, editors, *Human Vision and Electronic Imaging III*, volume 3299 of *Proceedings of SPIE*, pages 232–249, San Jose, California, January 1998.

Marc Proesmans, Luc J. Van Gool, Eric J. Pauwels, and André Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In J.-O. Eklundh, editor, *Proceedings of the Third European Conference on Computer Vision*, volume 801 of *Lecture Notes in Computer Science*, pages 295–304, 1994. doi:10.1007/BFb0028329.

Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian conditional random fields. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 269–276, Barbados, January 2005.

Carl E. Rasmussen. `minimize.m` - Conjugate gradient minimization. `http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/`, September 2006.

Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing*. Academic Press, 2nd edition, 1982.

Bodo Rosenhahn, Thomas Brox, and Joachim Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, 73(3):243–262, July 2006. doi:10.1007/s11263-006-9965-3.

Michael G. Ross and Leslie Pack Kaelbling. Learning static object segmentation from motion segmentation. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 956–961, Menlo Park, California, 2005. AAAI Press.

Stefan Roth. Analysis of a deterministic annealing method for graph matching and quadratic assignment problems in computer vision. Diplom thesis, University of Mannheim, Germany, May 2001.

Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, San Diego, California, June 2005a. doi:10.1109/CVPR.2005.160.

Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 1, pages 42–49, Beijing, China, October 2005b. doi:10.1109/ICCV.2005.180.

Stefan Roth and Michael J. Black. Specular flow and the recovery of surface structure. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1869–1876, New York, New York, June 2006. doi:10.1109/CVPR.2006.290.

Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50, August 2007. doi:10.1007/s11263-006-0016-x.

Stefan Roth, Fulvio Domini, and Michael J. Black. Specular flow and the perception of surface reflectance. *Journal of Vision*, 3(9):413a, 2003. doi:10.1167/3.9.413.

Stefan Roth, Leonid Sigal, and Michael J. Black. Gibbs likelihoods for Bayesian tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, Washington, DC, June 2004. doi:10.1109/CVPR.2004.116.

Carsten Rother. Personal communication, 2007.

Daniel L. Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, November 1994. doi:10.1088/0954-898X/5/4/006.

Daniel L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37(23):3385–3398, December 1997. doi:10.1016/S0042-6989(97)00008-4.

Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, November 1992. doi:10.1016/0167-2789(92)90242-F.

Phil Sallee and Bruno A. Olshausen. Learning sparse multiscale image representations. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1327–1334, 2003.

Hanno Scharr. Optimal filters for extended optical flow. In *First International Workshop on Complex Motion*, volume 3417 of *Lecture Notes in Computer Science*. Springer, 2004.

Hanno Scharr and Hagen Spies. Accurate optical flow in noisy image sequences using flow adapted anisotropic diffusion. *Signal Processing: Image Communication*, 20(6):537–553, July 2005. doi:10.1016/j.image.2005.03.005.

Hanno Scharr and Joachim Weickert. An anisotropic diffusion algorithm with optimized rotation invariance. In G. Sommer, N. Krüger, and C. Perwass, editors, *Pattern Recognition, Proceedings of the 22nd DAGM-Symposium*, pages 460–467. Springer, 2000.

Hanno Scharr, Michael J. Black, and Horst W. Haussecker. Image statistics and anisotropic diffusion. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, pages 840–847, Nice, France, October 2003. doi:10.1109/ICCV.2003.1238435.

Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3): 7–42, April 2001. doi:10.1023/A:1014573219977.

Christian Schellewald, Stefan Roth, and Christoph Schnörr. Evaluation of convex optimization techniques for the weighted graph-matching problem in computer vision. In B. Radig and S. Florczyk, editors, *Pattern Recognition, Proceedings of the 23rd DAGM-Symposium*, volume 2191 of *Lecture Notes in Computer Science*, pages 361–368. Springer, 2001.

Christian Schellewald, Stefan Roth, and Christoph Schnörr. Performance evaluation of a convex relaxation approach to the quadratic assignment of relational object views. Technical Report TR-2002-02, University of Mannheim, Germany, February 2002.

Christian Schellewald, Stefan Roth, and Christoph Schnörr. Evaluation of a convex relaxation to a quadratic assignment matching approach for relational object views. *Image and Vision Computing*, 2007. doi:10.1016/j.imavis.2006.08.005. To appear.

Christoph Schnörr. Unique reconstruction of piecewise-smooth images by minimizing strictly convex nonquadratic functionals. *Journal of Mathematical Imaging and Vision*, 4(2):189–198, May 1994. doi:10.1007/BF01249896.

Christoph Schnörr, Rainer Sprengel, and Bernd Neumann. A variational approach to the design of early vision algorithms. *Computing Supplement*, 11:149–165, 1996.

Giovanni Sebastiani and Fred Godtliebsen. On the use of Gibbs priors for Bayesian image restoration. *Signal Processing*, 56(1):111–118, January 1997. doi:10.1016/S0165-1684(97)00002-9.

Solomon E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410, August 1994. doi:10.1016/0004-3702(94)90072-8.

Leonid Sigal, Sidharth Bhatia, Stefan Roth, Michael J. Black, and Michael Isard. Tracking loose-limbed people. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 421–428, Washington, DC, June 2004. doi:10.1109/CVPR.2004.252.

S. D. Silvey. *Statistical Inference*. Chapman & Hall, 1975.

Eero P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. In P. Müller and B. Vidakovic, editors, *Bayesian Inference in Wavelet Based Models*, volume 141 of *Lecture Notes in Statistics*, pages 292–308. Springer, 1999.

Eero P. Simoncelli. Statistical modeling of photographic images. In A. Bovik, editor, *Handbook of Video and Image Processing*, chapter 4.7. Academic Press, 2nd edition, 2005.

Eero P. Simoncelli, Edward H. Adelson, and David J. Heeger. Probability distributions of optical flow. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 310–315, Lahaina, Maui, Hawaii, June 1991. doi:10.1109/CVPR.1991.139707.

A[nuj] Srivastava, A[nn] B. Lee, E[ero] P. Simoncelli, and S[ong]-C[hun] Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, January 2003. doi:10.1023/A:1021889010444.

Jean-Luc Starck, Emmanuel J. Candès, and David L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, June 2002. doi:10.1109/TIP.2002.1014998.

Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrázek, and Martin Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686–713, January 2004. doi:10.1137/S0036142903422429.

Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. Technical Report P-2551, MIT Laboratory for Information and Decision Systems, Cambridge, Massachusetts, October 2002.

Deqing Sun and Wai-Kuen Cham. An effective postprocessing method for low bit rate block DCT coded images. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, Hawaii, April 2007.

Jian Sun, Nan-Ning Zhen, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, July 2003. doi:10.1109/TPAMI.2003.1206509.

Richard Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301, December 1990. doi:10.1007/BF00126502.

Rick Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for Markov random fields. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proceedings of the Ninth European Conference on Computer Vision*, volume 3952 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2006. doi:10.1007/11744047_2.

Marshall F. Tappen and William T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, pages 900–907, Nice, France, October 2003. doi:10.1109/ICCV.2003.1238444.

Marshall F. Tappen, Brian C. Russell, and William T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Proceedings of the 3rd International Workshop on Statistical and Computational Theories of Vision*, Nice, France, October 2003.

Yeh Whye Teh, Max Welling, Simon Osindero, and Geoffrey E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, December 2003.

Demetri Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, July 1986.

A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

Michael E. Tipping and Christopher M. Bishop. Bayesian image super-resolution. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1279–1286, 2003.

Håkon Tjelmeland and Julian Besag. Markov random fields with higher-order interactions. *Scandinavian Journal of Statistics*, 25(3):415–433, September 1998. doi:10.1111/1467-9469.00113.

C[arlo] Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, Bombay, India, January 1998. doi:10.1109/ICCV.1998.710815.

Philip H. S. Torr. Personal communication, 2007.

Antonio Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, July 2003. doi:10.1023/A:1023052124951.

Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(2):391–412, August 2003. doi:10.1088/0954-898X/14/3/302.

Yanghai Tsin, Visvanathan Ramesh, and Takeo Kanade. Statistical calibration of CCD imaging process. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 1, pages 480–487, Vancouver, British Columbia, Canada, July 2001. doi:10.1109/ICCV.2001.937555.

Jun Tsuzurugi and Masato Okada. Statistical mechanics of the Bayesian image restoration under spatially correlated noise. *Physical Review E*, 66(6):066704, December 2002. doi:10.1103/PhysRevE.66.066704.

Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):657–673, May 2002. doi:10.1109/34.1000239.

Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 1, pages 18–25, Nice, France, October 2003. doi:10.1109/ICCV.2003.1238309.

J. H. van Hateren and D[aniel] L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 265 (1412):2315–2320, December 1998. doi:10.1098/rspb.1998.0577.

J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 265(1394):359–366, March 1997.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, Department of Statistics, University of California, Berkeley, September 2003.

Martin J. Wainwright and Eero P. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 855–861, 2000.

Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. MAP estimation via agreement on (hyper)trees: Message passing and linear programming approaches. *IEEE Transactions on Information Theory*, 51(11):3697–3717, November 2005. doi:10.1109/TIT.2005.856938.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. doi:10.1109/TIP.2003.819861.

Joachim Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing Supplement*, 11:221–236, 1996.

Joachim Weickert. A review of nonlinear diffusion filtering. In *Proceedings of Scale-Space Theory in Computer Vision*, volume 1252 of *Lecture Notes in Computer Science*, pages 3–28, Berlin, Germany, 1997. Springer.

Joachim Weickert. Applications of nonlinear diffusion in image processing and computer vision. In *Proceedings of Algorithmy 2000*, volume LXX of *Acta Math. Univ. Comenianae*, pages 33–50, 2001.

Joachim Weickert and Christoph Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3): 245–255, May 2001. doi:10.1023/A:1011286029287.

Yair Weiss. Belief propagation and revision in networks with loops. AI Memo 1616, Massachusetts Institute of Technology, Cambridge, Massachusetts, November 1997.

Yair Weiss and Edward H. Adelson. Slow and smooth: A Bayesian theory for the combination of local motion signals in human vision. AI Memo 1624, Massachusetts Institute of Technology, Cambridge, Massachusetts, February 1998.

Yair Weiss and William T. Freeman. Personal communication, 2007a.

Yair Weiss and William T. Freeman. What makes a good model of natural images? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, June 2007b. To appear.

Max Welling and Sridevi Parise. Bayesian random fields: The Bethe-Laplace approximation. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, Cambridge, Massachusetts, July 2006.

Max Welling, Geoffrey E. Hinton, and Simon Osindero. Learning sparse topographic representations with products of Student-t distributions. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1359–1366, 2003.

Heiko Wersing, Julian Eggert, and Edgar Körner. Sparse coding with invariance constraints. In *Proceedings of the 13th International Conference on Artificial Neural Networks*, pages 385–392, Istanbul, Turkey, June 2003.

Christopher K. I. Williams. How to pretend that correlated variables are independent by using difference observations. *Neural Computation*, 17(1):1–6, January 2005. doi:10.1162/0899766052530884.

Christopher K. I. Williams and Felix V. Agakov. An analysis of contrastive divergence learning in Gaussian Boltzmann machines. Technical Report EDI-INF-RR-0120, University of Edinburgh, UK, May 2002.

Alan S. Willsky. Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, August 2002. doi:10.1109/JPROC.2002.800717.

E. Wong. Two-dimensional random fields and representation of images. *SIAM Journal on Applied Mathematics*, 16(4):756–770, 1968.

Frank Wood, Stefan Roth, and Michael J. Black. Modeling neural population spiking activity with Gibbs distributions. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1539–1546, 2006.

O[liver] J. Woodford, I[an] D. Reid, P[hilip] H. S. Torr, and A[ndrew] W. Fitzgibbon. Fields of experts for image-based rendering. In *Proceedings of the British Machine Vision Conference*, volume III, page 1109ff, Edinburgh, UK, September 2006.

John W. Woods. Two-dimensional discrete Markovian fields. *IEEE Transactions on Information Theory*, 18(2):232–240, March 1972.

Chen Yanover, Talya Meltzer, and Yair Weiss. Linear programming relaxations and belief propagation – An empirical study. *Journal of Machine Learning Research*, 7:1887–1907, September 2006.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–236. Morgan Kaufmann Pub., 2003.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, July 2005. doi:10.1109/TIT.2005.850085.

Alan L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, July 2002. doi:10.1162/08997660260028674.

Alan [L.] Yuille. The convergence of contrastive divergences. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 1593–1600, 2005.

Alexey Zalesny and Luc van Gool. A compact model for viewpoint dependent texture synthesis. In *Proceedings of SMILE 2000 Workshop*, volume 2018 of *Lecture Notes in Computer Science*, pages 124–143, 2001.

Song Chun Zhu and Xiuwen Liu. Learning in Gibbsian fields: How fast and accurate can it be? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):1001–1006, July 2002. doi:10.1109/TPAMI.2002.1017626.

Song Chun Zhu and David Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, November 1997. doi:10.1109/34.632983.

Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, November 1997. doi:10.1162/neco.1997.9.8.1627.

Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, March 1998. doi:10.1023/A:1007925832420.