Efficient Statistical Methods for 3D Shape Inference

Brian Potetz

CMU-CS-08-128

April 25, 2008

School of Computer Science
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Tai Sing Lee, CMU, Chair
John Lafferty, CMU
Ann Lee, CMU
Mike Lewicki, CMU
Alan Yuille, UCLA

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Visual inference is a complex and ambiguous problem, and these properties have presented a significant obstacle to developing effective algorithms for many visual tasks. In this thesis, I begin by developing a methodology for statistical inference that is particularly suited for the complex tasks of visual perception. The approach is based on Belief Propagation, a highly successful inference technique that has lead to notable progress in a number of statistical inference applications. Unfortunately, the computational complexity of belief propagation allows it to be applied to only fairly simple statistical distributions, thus excluding many of the rich statistical problems encountered in computer vision. In this thesis, I introduce a new technique to reduce the computational complexity of belief propagation from exponential to linear in the clique size of the underlying graphical model. These advancements allow us to efficiently solve inference problems that were previously intractable.

I then apply this methodology to several visual tasks. In one example, I develop a statistical approach to the problem of estimating 3D shape from shading in a single image, a classic problem of computer vision that has been a subject of research since the lunar surface studies of the 1920's. Previous approaches typically have worked by forming a deterministic model of image formation and then attempting to invert this model. These approaches struggled with the nonlinearity and ambiguity inherent in the problem; the best algorithms were described as "generally poor" in a recent survey [109]. The statistical approach introduced here produces fairly convincing reconstructions, and also offers several novel flexibilities that previous approaches lack.

One difficulty faced by shape-from-shading and several other areas of computer vision is the ambiguity inherent in the problem. To produce successful statistical inference in an underconstrained problem, we must exploit a strong statistical prior. While previous applications of belief propagation could only be run using weaker, pairwise-connected models of spatial priors, the efficient techniques introduced here make more sophisticated approaches possible. Additionally, I address the issue of learning the parameters of spatial priors, by leveraging the power of efficient belief propagation towards efficient learning. These learned spatial priors are then applied successfully to image denoising and shape-from-shading.

Contents

List of Figures

List of Tables

Acknowledgements

I first and foremost must thank my advisor, Tai Sing Lee. His diverse interests and expertise have provided me with endless exciting opportunities and a versatile background that would otherwise have required ten advisors to acquire. Beyond this, I am grateful for his generosity as a mentor, both in his perpetual guidance and commitment, and in allowing me the freedom to forge some directions of my own. I have been very fortunate to work with Tai Sing, and I will strive to emulate his advising style with my own students in the future.

In addition, I am indebted to my thesis committee, for their thoughtful questions and helpful comments. I thank my labmates Ryan Kelly, Tom Stepleton, and Matt Smith for their generous help and support, and especially Jason Samonds, for his insightful collaboration and neuroscience expertise. I also thank all those who helped me while collecting range data: Frank Li, Kui Shen, Scott Marmer, and especially Mark Albert, who sacrificed many sunny afternoons and predawn hours to scrambling around Pittsburgh in search of perfect range data.

I also thank the faculty, students, and administrators that have helped make CMU the exciting, collegial, and wondrous learning and research environment that it is. Attending CMU and contributing to the amazing research that goes on here has been a dream of mine since visiting as a child, while my sister was choosing a college to attend. It has lived up to these wild expectations.

I thank my parents, for all their encouragement, and for teaching me the beauty in science.

Finally, I thank my wife, Sarah, for her undying support and understanding. This thesis is dedicated to her.

This research was sponsored by the National Science Foundation Graduate Research Fellowship Program, National Institute on Drug Abuse under grant no. 1T90DA02276201, and the National Science Foundation under training grant DGE-9987588. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Chapter 1

Introduction

In the past decade, one of the greatest lessons learned by the artificial intelligence community is that when dealing with a complex and uncertain universe, statistical approaches have a strong advantage. By being aware of the variation of expected outcomes, statistical solutions to artificial intelligence problems are more robust in real-world situations. In the field of computer vision specifically, statistical approaches such as Bayesian inference, markov random fields, and particle filtering have lead to significant progress. In spite of these advances, statistical approaches to the inference of depth in single images are still in their infancy. In fact, little is known about the joint statistics of natural 3D shape and natural images.

The inference of 3D shape from single images has been a topic of serious research for many decades. While many approaches have been suggested, the great majority of these approaches have been based on physical models of the interaction between light and surfaces. For instance, the problem of inferring shape from surface shading is typically approached by starting with a deterministic mathematical model of image formation, and then trying to invert this model. Unfortunately, inverting the image formation process is highly underconstrained. This forces us to revert to oversimplified models of image formation which may be unrealistic in natural scenes. Various assumptions about image formation parameters have to be made, such as Lambertian surface reflectance, uniform albedo, and shadow-free, single point source illumination. However, these assumptions are often violated in the real world, and this leads to poor generalization for these algorithms.

Statistical approaches to inferring shape need not be constrained to a single, simplified physical model of image formation. Instead, such an approach will be aware of a distribution of shapes that could have resulted in a particular image under a variety of likely lighting and reflectance conditions. Rather than assuming unknown parameters to hold some typical value (like assuming reflectance to be Lambertian, or albedo to be constant), a statistical approach can marginalize across unknown parameters to deduce the most likely shape under unknown conditions.

Another advantage of statistical approaches to shape inference is that natural scenes may contain exploitable statistical regularities that are not readily apparent from physical models of image formation. The natural world contains many unexplored statistical properties, such as the natural geometry of objects, distributions of the size and number of objects, the arrangements of objects in space, regularities in the position of the observer, natural distributions of light, and the statistics of surface properties of those objects. All of these properties may have regularities in the real world that could be exploited by statistical inference algorithms. One simple example of such a regularity is the tendency for light to come from above. This regularity is highly exploited by the human visual system [72].

Additionally, methods of depth inference that invert physical models of image formation are typically forced to neglect image formation phenomena that are mathematically cumbersome or difficult to invert. Examples include cast shadows, diffuse lighting, interfacet reflection, and subsurface light transport. Each of these phenomena are important in computer graphics for rendering photorealistic images, and omitting them from a depth inference algorithm invariable results in bias when that algorithm is applied to real natural scenes. Perhaps more importantly, another fundamental drawback of omitting these phenomena is that, while each of these aspects of image formation is mathematically complex within a deterministic model, when these phenomena are considered together they can result in robust statistical trends that can be exploited for improved depth inference. In other words, image formation phenomena that are too complex mathematically to incorporate in a deterministic framework can actually result in stochastic depth cues that are *beneficial* to statistical depth inference techniques. One of my earlier studies shows an example of this. Using a database of laser-acquired range images, we have found that darker pixels tend to be farther away than bright pixels [67]. Evidence suggested that this is due to the effect of shadowing in complex natural scenes. Specifically, concavities and visible interiors of complex objects are more likely to be in shadow. For this effect to be significant, complex non-smooth 3D shapes, cast shadows, diffuse lighting, and interfacet reflection must all be present within the scene. Each of these are aspects of image formation that are most often ignored by previous depth inference approaches, and yet, together they produce helpful stochastic depth cues that can be exploited in a statistical setting. Later we showed that, if restricted to linear relationships between depth and intensity, these simple shadow cues are often more powerful than shading cues in natural scenes for inferring high-resolution 3D shape from single images [68]. This type of cue was originally predicted to exist only for aerial photography scenes (viewed from directly above) under cloudy conditions [52]. It has been the subject of very few investigations. Shape from shading (SFS), on the other hand, has been studied extensively. And yet, we were able to show that the more obscure shadow cue was actually the stronger cue in a database of outdoor images, all acquired under sunny conditions, and all taken with the camera pointing towards the horizon. This type of discovery suggests that studying the natural statistics of scenes is likely to reveal some surprising or even counter-intuitive trends that would be difficult to predict theoretically using physics-based models of image formation.

A fourth major advantage of developing Bayesian shape inference algorithms for single images is that they can be readily integrated with other sources of depth cues, such as multiple images (including stereo images and motion), or high-level knowledge of the environment. Consider stereo cues. Stereo vision is a powerful depth cue, and algorithms that extract depth from stereo have become fairly successful. However, there are certain weaknesses intrinsic to all stereo algorithms; weaknesses that monocular cues may help to clarify. First, because image disparity is inversely proportional to distance, the strength of the stereo cue diminishes quickly with distance [15]. Monocular cues such as shading do not suffer from this limitation. Also, solving the stereo correspondence problem requires some trade-off between the ability to confidently match two corresponding points in the stereo pair, and the size of the image region used to characterize each point [14, 77]. Because of this, shape estimates from stereo are often accurate in the low spatial frequencies, but fail at inferring the fine, high frequency details of shape. Shading cues, on the other hand, are still powerful in the high spatial frequencies. Because of their naturally complementary qualities, there have been many attempts to integrate stereo and shape from shading algorithms. However, this is still an open problem. As we will discuss later on, Bayesian inference provides a natural framework for the integration of these two cues.

In addition to advancing algorithms to infer depth, a greater understanding of natural scene statistics and statistical approaches to depth inference might lead to insight into the way the human visual system solves these same problems. The human brain evolved and developed under natural statistical conditions, without any access to analytical, physical laws of light interaction. Therefore, an understanding of statistical approaches to 3D shape inference is of great relevance to uncovering the processes in the brain responsible for surface inference. This approach has been advocated before. The notion that human perception and behavior was best understood in the context of our natural environment was advanced by Gibson in the 1960's [25]. Since that time, developments in the statistics of natural scenes have been highly useful for understanding human perception in both psychology and in neuroscience. By studying the statistics of natural images, researchers have developed ways to process and encode natural images so as to maximize efficiency, either in a metabolic sense, or in terms of information transmission. Neuroscientists then showed that many of the basic properties of retinal ganglion cells and V1 cortical neurons can be understood in terms of representing images efficiently [17]. These insights are a promising beginning in unraveling how the brain processes visual stimuli.

However, storing images efficiently is not an end-goal of the visual system. Efficient encoding is only a subgoal that may be useful in accomplishing the brain's many visual tasks. The true purpose of the visual system is to infer the underlying properties of an image: for example recognizing objects, inferring depth, or estimating materials. Studying the statistics of images alone cannot help us to understand how the brain accomplishes these goals. In order to study these visual tasks in the context of the natural environment, we will need to study both natural images and the ground-truth values of the underlying scene properties. The inference of depth is a good first choice for several reasons. First, natural range-images can be acquired by laser-scanner. In contrast, many mid-level visual tasks have computational goals that are more subjective in nature, such as scene segmentation, edge detection, or contour completion. The availability of ground-truth 3D data facilitates both meaningful statistical studies and the development and benchmarking of inference algorithms. Another benefit to studying depth inference in the brain is that signals corresponding to early forms of stereopsis (detecting matches between the left and right eyes) occur early in the visual stream and are fairly well understood. Having some early neural correlate of depth might prove very helpful for neurophysiological studies designed to better understand the 3D spatial priors exploited by the brain [75], or to study how multiple depth cues can be integrated [69].

The idea of the brain as an inference engine can be traced back to Helmholtz's unconscious inference theory of perception. More recently, as statistical methods in machine learning advance, Bayesian inference has been suggested as a general computational principle of the brain [56], or advanced as a possible method for solving several visual tasks [82]. We hope that the methods of statistical inference that we develop may be informative for understanding what types of computations may take place in the brain.

Chapter 2

Related Work

In this thesis, I develop a set of tools for statistical inference that are designed to benefit many branches of computer vision, and extend to any application that seeks to infer a value in a high-dimensional continuous space (such as the space of images) given a complex and tightly coupled statistical distribution. The applications I will examine here will surround the problem of monocular depth inference, and in particular, the inference of shape from shading. One goal of my research is to develop a flexible framework for visual inference that will help to generalize shape-from-shading to apply to broader and less restrictive classes of scenes. In this section, I review previous approaches to depth inference, and in particular approaches that are statistical or probabilistic in nature. I will begin with a look at studies of the statistics of 3D shape.

2.1 Statistics of natural 2D images

Psychologists and neuroscientists (such as Hermann von Helmholtz, Horace Barlow, and J.J. Gibson) have speculated throughout the past century about the importance of natural scene statistics for human perception. However, it was not until the late 1980's that researchers began to map out the basic statistical properties of natural images. Principal among these early studies was the discovery of the scale invariance of natural images [22, 74]. This means that images tend to have similar statistical properties when viewed at different scales. One such property the power spectrum, which scale invariance predicts should take the form $1/f^2$, where f is spatial frequency.

This prediction is fairly robust in single images, and highly robust when estimated over large image ensembles. Other statistical properties also obey laws of scale invariance, including the histograms of contrast within small patches [74], the distributions of various linear filters [54]. The property of scale invariance carries with it a whole family of statistical regularities. Such statistical regularities help us to recognize what images are surprising or salient, what images "should" look like (helpful when inferring obscured or unseen regions of an image), and what types of images should our visual systems be optimized for.

Another key discovery in the statistics of natural images is the prominence of edges in natural images. One early observations was that histograms of the response of linear filter tend to have highly kurtotic distributions, with sharp peaks and heavy tails [38]. Independent component analysis revealed edge-like gabor filters to be the most atomic elements of natural images [4]. Also, the joint distributions of neighboring wavelet coefficients were found to be sharply stellated in a way that suggested the prevalence of edge features such as extended edges and T-junctions [38].

One model that explains the bulk of these findings is the occlusion-based "collage" model of natural images [54]. In this model, images are approximated by collections of piecewise-constant regions that occlude one another. Sample images drawn from the collage model are constructed by repeatedly dropping opaque shapes at random locations on the image, with the size of each shape drawn from a certain distribution $f(r)$. If $f(r) = 1/r^3$, the resulting images can be shown to be scale invariant.

2.2 Statistics of natural range images

Less is known about the statistics of natural range images. In the 1990's, the popularity of fractal models of natural phenomena lead researchers to measure the fractal dimension of natural 3D range scans. These scans typically had fractal dimensions ranging from 2.0 to 2.6 (corresponding to power spectra drop-off rates of $1/f^{2.0}$ to $1/f^{2.8}$ respectively) [1].

Later, a more complete study of the statistics of range images was performed [37]. This study found that range images obey many laws of scale invariance. It was also found that many of the statistical properties of resembled those of natural images, and could also be explained by the collage model. A third study of range image statistics found that the sizes of planar objects obeys a $1/r^{2.4}$ power-law, where r is the object radius [102]. This finding may be related to the collage model, which predicts that the sizes of objects in the visual plane, including any occluded portion, should follow a $1/r^3$ distribution.

2.3 Statistics of natural 3D scenes

Even less is known about the joint statistics of natural range and color images. The technology required to simultaneously acquire ground-truth range images and coregistered color images has only recently become readily available. There are currently three databases of natural coregistered images and range images being used for scientific study. One of these was collected by Dale Purves [36], using a Riegl LMS-Z360 range scanner. This group focuses primarily on explaining psychophysical phenomenon through range image statistics. Although their has produced some results on the statistics of range images [102], their work typically does not make use of the luminance component of the database, or study the statistical relationship between light and depth.

Another such database was collected in 2005 by Andrew Ng [76]. This database has high-resolution color images (1704×2272), but only low-resolution range data (86×107). This database was collected using a SICK 1D laser range scanner, mounted on a motor to collect a 2D array of range scans. Their SICK scanner had an operational range of 81 meters, as compared with the 300 meter range of the Riegl LMS-Z360. Because the color camera was a separate unit, they report some alignment errors between the range and color data (± 2 range units). Their work has focused on the inference of depth from monocular images, specifically for obstacle detection in autonomous vehicles. Monocular inference relates directly to our second aim, and we discuss their work in more detail in the following section. However, their work emphasizes performing inference without understanding the underlying statistics, and so they have no results that model, explain, or analyze the statistics of natural scenes directly.

Our own database was collected in June 2002, also using a Riegl LMS-Z360 range scanner. We used this database in a series of studies to identify statistical trends that exist in real 3D scenes, and which may provide insight into how depth can be inferred from monocular cues, both in computer vision and in the brain. One major finding from this work was the discovery that

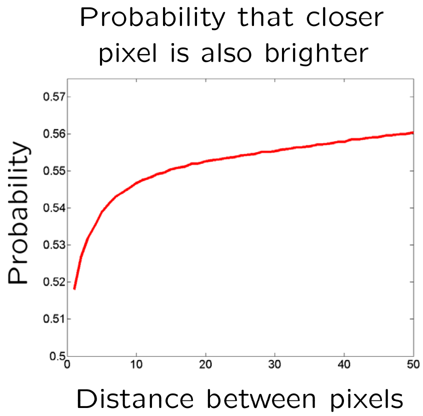


Figure 2.1: Using a database of coregistered range and color images, we showed that brighter image regions are more likely to be closer to the observer than dark regions. Given two pixels spaced sufficiently far apart, an observer who guessing that the brighter pixel is the nearer of the two will be right more than 56% of the time. This effect is due to the effect of shadows in complex natural shapes, which often contain concavities and object interiors.

pixel distance and pixel darkness were significantly correlated ($\rho = 0.23$). In [67], I demonstrated that this statistical effect is due to cast shadows: image regions that lie within concavities and surface interiors are more likely to lie in shadow, and they are also more likely to be farther from the observer than points on object exteriors. This effect is especially powerful in

- foliage, where areas deeper and further into a tree or wooded area are also darker due to shadowing
- piles of objects, where crevices between objects lie in shadow
- folds of fabric or other materials, where fold interiors lie in shadow

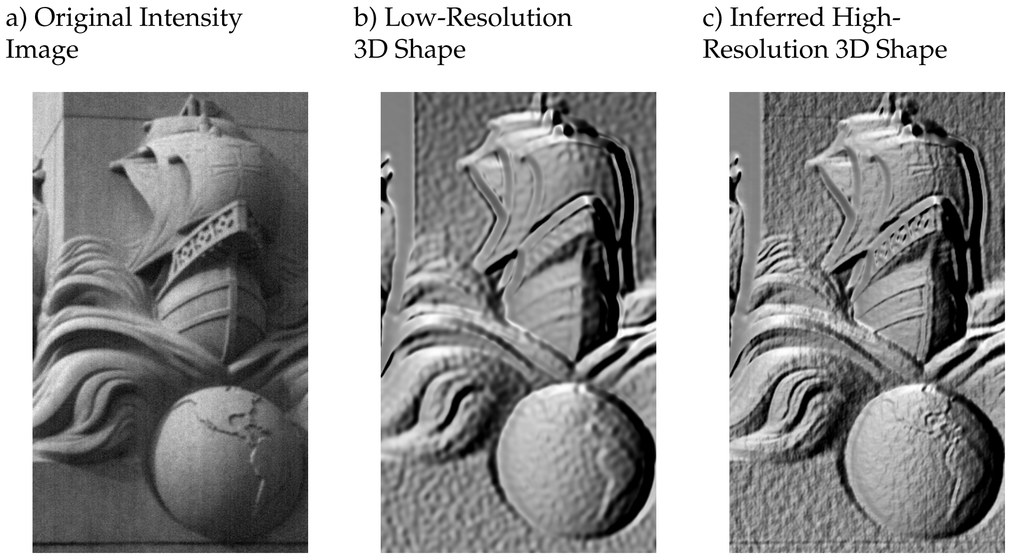
These same principles repeat often enough in nature to produce a small but robust correlation between closeness and brightness. Figure 2.1 shows the extent to which bright pixels tend to lie nearer the observer for outdoor, sunlit natural scenes.

This finding serves as one example of how a statistical study can uncover trends in real scenes that may not be immediately obvious by studying mathematical models of image formation, and yet can be exploited to achieve better depth inference. A correlation between nearness and brightness has been predicted using mathematical models, but only for aerial photography taken under perfectly diffuse lighting conditions [52]. Our results show that the effect remains strong even for sunlit conditions under oblique lighting. This result is much more difficult to model mathematically. Without statistical investigations into these effects, the strength of the nearness/brightness correlation for natural, non-aerial images would likely have been underestimated.

In addition, the nearness/brightness correlation shows that some relationships between images and their underlying 3D shapes may be difficult to exploit in a deterministic setting, but quite simple in a non-deterministic, or statistical inference approach. In order for the nearness/brightness correlation to be strong in natural outdoor scenes, complex 3D shapes, shadowing, diffuse lighting, and inter-facet reflection must play a role. All of these phenomena are very cumbersome mathematically, and deterministic models of image formation that include these phenomena are highly difficult, or likely even intractable to invert. Together, however, these image formation effects combine to produce a statistical relationship that is maximally simple: an absolute correlation between brightness and nearness. For a statistical approach to depth inference, such a depth cue should be trivial to exploit.

Later, we extended our analysis of the first order statistics of image/range image pairs by studying how the relationship between shape and appearance changes over scale [68]. A careful characterization of these statistical properties extended naturally to an algorithm to infer high-resolution 3D shape from a sparse, low-resolution depth map and a full-resolution color image (see figure 2.2). Such algorithms have applications in medical imaging and robotic navigation. Using the low-resolution range data, we can learn the monocular cues from the low-resolution data, and then extrapolate that relationship into the higher spatial frequencies using our statistical model. The resulting algorithm not only achieved state-of-the-art performance, but provided additional insight into the statistics of natural scenes. Our analysis revealed that the majority of the algorithm performance was due to shadow cues, while shading cues were of secondary importance. This discovery was surprising to many in the field; shading has received much greater attention in the past than shadow-based cues.

In addition to acting as a possible depth cue for 3D shape inference, our discovery of the nearness/brightness correlation in natural scenes also provided insight into how depth is computed in the brain. This statistical finding



a) Original Intensity Image

b) Low-Resolution 3D Shape

c) Inferred High-Resolution 3D Shape

Figure 2.2: **a)** An example image from our database. **b)** The corresponding range image was subsampled to produce a low-resolution depth map, and then (for illustration purposes) rendered to create an artificial, computer-generated image. Next, a computer algorithm was used to learn the statistical relationship between the low-resolution 3D shape of **b)** and the 2D image of **a)**. This includes both shading and shadow (nearness/brightness correlation) cues. In this example, shadow cues were stronger. This learned statistical relationship was then extrapolated into higher spatial frequencies to estimate the high-resolution 3D shape, shown in **c)**.

provided an ecological explanation to a psychophysical phenomenon that had been known since the time of Leonardo da Vinci, who stated that *"Among bodies equal in size and distance, that which shines the more brightly seems to the eye nearer"*. Later, psychophysicists validated da Vinci's observations in rigorous, controlled experiments, [2, 8, 86, 13, 20, 83, 91, 53, 100] where the effect was sometimes referred to as the depth cue of *relative brightness*. Previous explanations for this perceptual effect were primarily psychological in nature, and explained relative brightness as an artifact of perception rather than as an adaptive behavior that exploits real properties the environment. The prevailing explanation was that brighter image regions appeared larger due to the irradiation of light (scattering in the atmosphere), and since larger objects were more likely to be near, brighter objects would be perceived as

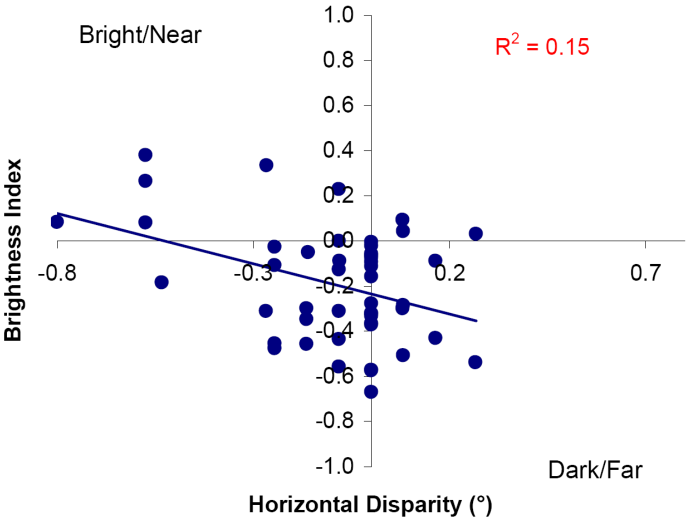


Figure 2.3: As predicted by our statistical studies, V1 neurons that prefer bright stimuli also tend to prefer near disparities. In this plot, the preferred brightness of each neuron is plotted against its preferred 3D disparity.

nearer [86, 13]. Our findings provided an adaptive, ecological explanation for relative brightness.

Given the simplicity of this statistical relationship, we expected that the visual system should learn and exploit this cue fairly easily. Since early areas of the visual cortex have access to both luminance signals and absolute depth signals (via the comparison of left and right eye signals), we expected to find this relationship encoded fairly early on in the visual stream, in areas V1 and V2. To test this hypothesis, we measured the luminance preferences and depth preferences of 48 V1 cells in awake behaving macaque monkeys. We found that a cell's preferred brightness was correlated with its preferred disparity (nearness) with a correlation coefficient of 0.39 ($p = 0.01$). See figure 2.3.

This result demonstrates how studying the statistics of images together with underlying, behaviorally relevant scene properties can be useful for understanding visual inference in the brain. It is worth noting that this is a case where an understanding of natural statistics motivated the discovery of a new neurophysiological phenomenon. This is unusual - while natural scene statistics has been highly useful for understanding brain function, typically natural image statistics have only been used to explain neural phenomenon that was already discovered. This discovery opens up a new avenue for exploring how multiple sources of information are combined in the brain, and inference is achieved by the visual system in the face of uncertainty.

2.4 Statistical Approaches to 3D Shape Inference

As mentioned above, most monocular shape inference algorithms in the past have worked by inverting simple models of image formation. However, there have been some notable exceptions. In the next few sections, we will differentiate between *probabilistic* and *statistical* approaches. We will refer to an approach as probabilistic if it constructs an explicit model of the posterior probability distribution $P(Z|I)$. A method is statistical if it learns this model from the statistics of natural scenes (although we will include methods that learn from computer generated scenes in this category). Many classical approaches to depth inference (shape from shading, stereo, etc) have been formulated probabilistically.

2.4.1 Probabilistic Approaches to Shape From Shading

The problem of shape from shading (SFS) is to recover the 3D surface shape given a single image, where it is assumed that:

- the scene is lit from a single light source
- the light source is infinitely far away
- the direction of illumination is known
- all surface materials are uniform in albedo (i.e. the scene is entirely painted white)
- all surfaces are Lambertian (matte) in reflectance

For Lambertian materials, the image intensity at a point is given by

$$i(x,y) = \max(0, \frac{1 + pp_s + qq_s}{\sqrt{1+p^2+q^2}\sqrt{1+p_s^2+q_s^2}}) \tag{2.1}$$

where $N = (p, q, 1)$ is the surface normal vector, and $S = (p_s, q_s, 1)$ is the known illumination vector. While it is trivial to compute light intensity from 3D shape, inverting equation 2.1 is difficult, partly because it is highly nonlinear, and partly because the problem is underconstrained. Specifically, for any given 2D image that satisfies the SFS constraints listed above, there is potentially a very large number of possible 3D surfaces that are consistent with that image. In other words, many 3D surfaces, when rendered according to equation 2.1 under the known lighting direction S will result in the same 2D image.

Most SFS algorithms are not designed to handle the ambiguity inherent in the problem. Typically, SFS algorithms require that the 3D shape of the surface is known along the image border, or they implicitly assume that the shape at the border is flat, or cyclic, or some other pre-assumed condition. Knowing the shape along the image border removes a great deal (and in some cases, all) of the ambiguity of SFS. When 3D shape along the image border is not known, the results are quite poor, even for the best algorithms [109]. The benefit of a probabilistic approach to SFS is that it allows us to resolve the ambiguity of the problem by using spatial priors to select the 3D shape that is maximally likely among those that are consistent with the image (according to equation 2.1). Unfortunately, the mathematical tools for solving probabilistic formulations of SFS were not previously available.

There have been a number of approaches to solving SFS. Propagation approaches work by spreading information from points with known surface normal (such as points where $i(x, y) = 1$) to surrounding regions. Energy minimization approaches formulate the SFS problem as a set of constraints that penalize poor reconstructions, and then attempt to minimize that energy function [55, 110]. Linear approaches work by inverting linear approximations to the Lambertian reflectance function [90]. More recently, a new class of solutions has emerged that computes minimum viscosity solutions of an eikonal equation that encodes the shape from shading problem. This approach only works in cases when the lighting direction matches, or nearly matches, the viewing direction (the lighting is from behind the camera). Surveys of these approaches to SFS are given by Zhang et. al. [109] and Durou et. al [19] (which includes minimum viscosity approaches).

Of these approaches, the energy minimization approach is the most relevant here, because they can be formulated probabilistically. Energy functions used for SFS problems typically have two components: an error term that penalizes surfaces that do not match the image, and a prior term that penalizes surface shapes judged to be unusual according to some heuristic, such as smoothness. These energy functions are easy to formulate as a statistical model of the posterior $P(Z|I) \propto P(I|Z)P(Z)$. In fact, on occasion, SFS energy functions are explicitly formulated this way [89, 40]. Unfortunately, the energy functions that are typical of this approach are difficult to minimize using traditional methods.

One interest in reviewing this literature is in determining how well these methods can be generalized to handle natural scenes. In natural scenes, multiple or non-constant albedos, unknown reflectance functions, and other complications will contribute to uncertainty in the distribution $P(N|i)$, where N is the surface normal, and i is the image intensity. Many energy minimization approaches to SFS are solved using methods that rely heavily on the deterministic nature of the reflectance function, or on its exact form. Also, the ambiguity of shading in natural scenes will require that many additional monocular depth cues be exploited. Methods of energy minimization that work well in simpler models of $P(Z|I)$ that capture only shading cues may not work well for more sophisticated models of the 3D shape posterior of natural scenes. For example, many variational SFS techniques seek minima of the energy functional by first applying the Euler equations from the calculus of variations. Researchers have had success with these methods only for very specific energy functionals; applying such methods to the more complex energy functionals of natural scenes would be discouragingly challenging from a mathematical standpoint. The energy minimization technique from classic SFS literature that shows the most potential for generalizing to inference in monocular natural scenes is the conjugate gradient descent method and its relatives. However, this method struggles with the local non-optimal minima of the traditional SFS formulation. I discuss such approaches to MAP estimation in section 3.2.

One particularly interesting application of gradient descent for statistical approach to shape from shading was made by Andre Jalobeanu [40]. This approach uses a graphical Bayes net to invert a standard, Lambertian model of image formation. Rather than learn a statistical model of the joint distribution of images and 3D surface shape, this algorithm uses a simple computer graphics rendering algorithm to artificially generate an image from a set of scene parameters (including surface shape, albedo, and lighting angle). For a given surface shape and coloring, they can estimate the likelihood that the observed image resulted from the hypothesized scene, $p(I|Z)$, by comparing the original image to the reconstructed image using a noise model. Using the Bayes rule and naturalistic priors of surface shape and albedo, the algorithm can then estimate the *a posteriori* probability, $p(Z|I)$. In order to improve the initial shape estimate, the derivative of error $(I - I_{est}[Z])^2$ with respect to shape Z is derived analytically, and then evaluated for successively improving shape estimates. While this approach has had some success with a restricted problem set (multiple viewpoint photographs of asteroid surfaces and aerial views of rural landscapes), it is not clear if this approach could work in natural scenes. Although this approach capitalizes on natural shape and reflectance priors, it does not fully benefit from the first two advantages of statistical inference of depth mentioned in the introduction. Like traditional shape from shading methods, it relies on the accuracy of one simplified physical model of image formation to perform inference. The rendering method used could be improved for greater accuracy. However, realistic rendering of natural scenes is highly complicated. Occlusion, interfacet reflection, secularities, and complex surface reflectance (for example, human skin is often modeled as several layers of translucent material) and other sophisticated techniques are required for accurate scene rendering. However, it is difficult to differentiate the error term for all but the most simplistic rendering models.

2.4.2 Learning from Computer Generated Images

In 1988, Lehky and Sejnowski used a 3-layer neural network to learn a relationship between an image and the orientations and magnitudes of principle curvatures of the underlying 3D shape [57]. The network was trained using the back-propagation algorithm on a database of Lambertian-rendered ellipsoids. The trained network performed well on scenes similar to the training set. Neural network approaches to the full classical SFS problem have also been attempted [5, 41]. A successful neural network approach to inferring depth from single, natural images would need to overcome several drawbacks of neural networks. Neural nets typically require a great deal of data to train. Also, once trained, it can be difficult to understand and debug the learned weights, which makes incremental progress difficult.

Knill and Kersten [42] describe a shape-from-shading technique where the Widrow-Huff learning rule is used to learn a linear function (linear in I) to approximate the MAP estimate $\hat{Z} = \max_Z P(Z|I)$. Linear approximations of the Lambertian reflectance function provide a reasonable estimate under oblique illumination conditions [64], and Knill's method has the benefit of a fractal prior. However, many depth cues are highly nonlinear, such as texture and perspective, making this technique difficult to generalize to natural scenes.

Another important example of learning shape-from-shading is VISTA (Vision by Image/Scene TrAining) [23]. VISTA has been applied to several problems, including a shape-from-shading problem where the goal is to factor an image into a “reflectance” image (where each pixel corresponds to surface albedo) and a range image. In general, VISTA uses a markov random field, defined on a grid, where each node corresponds to a small square patch of the scene. In the case of the SFS application, each node describes a patch from both the reflectance and range images. Thus, each node is a $2n^2$ dimensional random vector. Two grids are used for the SFS example - one whose nodes correspond to 8×8 patches, and one with 16×16 scene patch nodes. The patches of adjacent nodes overlap by one or two pixels. This allows VISTA to define the prior probability of a given scene by the degree to which neighboring patches agree on their regions of overlap:

$$P(z,r) = \prod_{i,j} exp(-|\vec{d_i} - \vec{d_j}|^2/2\sigma^2) \tag{2.2}$$

where $\vec{d_i}$ is the region of overlap with patch j. VISTA differs from most other MRF approaches in that instead of defining an explicit likelihood $P(z,r|i)$, VISTA uses a database of observed natural scenes. All range and reflectance patches must come from this database of examples (called “exemplars”). The likelihood is then defined by the agreement between the observed image patch and the exemplar:

$$P(z,r|i) = \prod_{patches} exp(-|\vec{I}_{exemplar} - \vec{I}_{observed}|^2/2\sigma^2) \tag{2.3}$$

Training VISTA consists of generating 200,000 training images with known range and reflectance images. Each training image is broken into 8×8 and 16×16 patches, and each patch is stored in a database. To infer shape and reflectance from a single image, each node first selects a set of 10 or 20 candidate exemplars whose image patches agree with the observed image patch. Loopy belief propagation is then used to select the configuration of candidate patches that maximize the posterior probability.

VISTA was trained on computer generated scenes. Each scene was either a flat surface painted with random ellipses, or a constant-albedo Lambertian surface formed by ellipsoidal bumps. All training and test images were lit from the same direction - the network must be retrained for any change in material or illumination parameters. In this simplified setting, the algorithm succeeded in distinguishing between all-reflectance or all-shape input images. However, the resulting range and reflectance images were often highly noisy. This is one of the biggest drawbacks to the VISTA method: the space of possible natural scenes is extremely large, and vast numbers of example patches would be required to adequately model it. This problem can only be expected to worsen for natural scenes.

One other weakness of VISTA is that the compatibility function between neighboring nodes is fairly weak. Bottom-up monocular depth cues can be highly ambiguous, especially if they are computed over a small spatial window. Rewarding candidate pairs that appear similar over a narrow strip of pixels is not strong enough to disambiguate these weak bottom-up cues.

2.4.3 Objects of Known Category

Finally, there is a wealth of literature dealing with 3D shape inference for items of known object category. This includes 3D face reconstruction [3], constructing 3D models of building from satellite images (see [79] for a review), and even ferns and trees [26]. The effectiveness of these methods typically lie in the strength of their prior models and in simplifying assumptions that are particular to their object category. Nevertheless, it is important to remain aware of these approaches.

2.4.4 3D Shape Inference From Multiple Images

It is also potentially useful to examine probabilistic approaches to inferring shape from *multiple* images. Bayesian approaches to stereo have proven quite successful, using loopy belief propagation on a markov random field grid [77, 82]. Similar probabilistic machinery has been used successfully for photometric stereo [84]. One important difference between these techniques and monocular shape inference is that, unlike monocular cues, both stereo and photometric stereo are highly local cues. Consider photometric stereo. In a shadowless Lambertian scene, just three images acquired under different, known illuminations are sufficient to completely determine the surface normal at each point. In natural images, where surfaces may be non-Lambertian or shadowed, the probability distribution over depth, $P(N(x,y)|i_1,\ldots,i_n)$, is still highly constrained (i.e., the distribution has low entropy). For multi-view stereo, the story is similar. Except within blank, textureless regions, likely values of depth $P(z(x, y)|i_1, \ldots, i_n)$ are highly constrained by the similarity of pixels at different disparities. For both stereo and photometric stereo, belief propagation serves primarily to incorporate a prior on 3D shape (such as smoothness) into the posterior distribution. For inference based on monocular depth cues, belief propagation will be required to play a much harder role. Even the most local monocular depth cues require significant nontrivial interaction among neighboring image regions. For instance, using traditional shape from shading methods, the intensity of a single pixel only restricts the surface normal to lie along a one-dimensional manifold of possibilities, even if the illumination properties, material reflectance, and albedo are all known. Significant interactions with neighboring pixels are required to output surface representations that are self-consistent (i.e. values of surface normals that integrate to a unique range image). This places much greater computational demands on the probabilistic machinery used to propagate local beliefs than was necessary in the multiple image case.

2.4.5 Statistical Inference of the Gist of the Scene and Coarse Shape Estimates

There have been a number of approaches that seek to infer the spatial "gist" of a scene, or the basic 3D structure of the scene as a whole [88, 63]. These works have shown that it is possible to infer basic spatial properties such as the mean absolute depth in one case [88], and subjective global 3D scene properties such as "openness", "expansion", "ruggedness", and "roughness" in another case [63]. The paper on the inference of mean absolute depth works by measuring the energy of wavelet responses at different locations within the image, and also the degree of correlation between wavelet responses across the image. The dimension of this feature space is then reduced using PCA, and a mixture of Gaussians is used to learn a probability distribution over the joint distribution $f(D, v)$, where D is absolute depth (as determined by a panel of impartial subjects for each image), and v is the feature vector of the image. Bayes rule and a model of the prior $p(v)$ is then used to obtain the expected absolute depth $E[D|v]$. The global 3D scene properties paper [63] uses similar methods. These works show that global 3D structure of a scene can be recovered using only a simple "spatial envelope" of the image.

More recently, a group from Stanford has taken this approach a step further [76]. Using similar texture-based cues, they infer a low-resolution (one twentieth scale) depth-map from a single high-resolution image. Specifically, 86×107 range images are inferred for each 1704×2272 color image. Their approach is similar to ours in that they use a database of real natural range scans and color images to train their model. We have described their database above (section 2.3). Their method divides each image into patches, and for each patch i they take a variety of texture-based statistics by summing over the square or absolute value of the output of filter F_n:

$$E_i(n) = \sum_{(x,y) \in patch(i)} |I(x,y) * F_n(x,y)|^k \tag{2.4}$$

for $k = 1, 2$. They use a portion of their database to train a set of parameters in a jointly Gaussian multiscale Markov random field (MRF). The MRF model includes two terms: a likelihood term that relates depth to the image features, and a smoothness term that acts as a prior on 3D shape. The likelihood term is $exp((d_i - x_i^T \theta)^2 / 2\sigma_1^2)$ where d_i is the depth at patch i, x_i is the feature vector from the image, and θ and σ_1 are parameters of the model. The smoothness term is $exp((d_i - d_j)^2 / 2\sigma_2^2)$, where i and j are neighboring patches, and σ_2 is a parameter of the model. Learning σ_2 allows the smoothness of the inferred 3D surface to depend on the features of the image. The model also has three scales, where depth at each scale is defined to be the average of depths at each included patch from the next smaller scale. The smoothness prior acts between neighboring patches at each depth. Using a joint gaussian distribution allows them to solve for d in closed form and compute the result very quickly. They also try a jointly Laplacian MRF, but this yields little additional improvement.

The algorithm was then tested on the portion of the range database not used for training. Average absolute-value error in the log-transformed range images was 0.132, which corresponds to a multiplicative factor of 1.36. In other words, a region with average log-error that is 10m away might be judged to be 13.6m or 7.38m away. This can be compared to baseline algorithm in which the output of the algorithm is always the mean range image (the average of all range images in the training portion of the database). The baseline algorithm has an average log error corresponding to a 1.97 multiplicative factor.

My approach to the depth inference problem differs from this in several important aspects. First, I am interested in inferring high-resolution surface

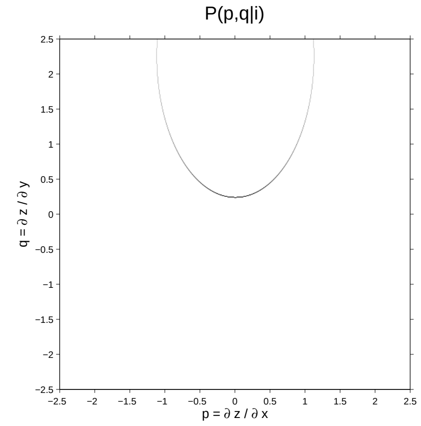


Figure 2.4: For a Lambertian surface with known albedo, the conditional joint distribution $P(p, q|i)$ is highly nongaussian - all non-impossible values lie along a 1D manifold. In the figure, lighting is from $(0, 1, 1)$, and pixel intensity $i = 0.85\rho$, where ρ is surface albedo. Dark values are more likely, white regions are impossible. The distribution pictured here assumes that all surface normal azimuths are equally likely.

data, including high-frequency 3D texture. Ultimately, this will require us to incorporate many monocular depth cues in addition to texture-based cues, including shading, shadow, perspective, and others. Multi-image cues, like stereo and motion, should also be easy to integrate into the model. Secondly, I do not expect that a jointly Gaussian or Laplacian model will be sufficiently effective in capturing the complex interdependencies between color images and surface shape, especially when non-texture-based cues are considered. For instance, given a Lambertian surface of constant albedo, if the lighting direction is known, the joint probability distribution between vertical slope q and horizontal slope p has all of its weight along a curved 1D manifold that corresponds to a level-set of the Lambertian equations (see figure 2.4). This distribution is highly non-Gaussian. Thirdly, I feel that a strong range prior is essential for successful inference of 3D surfaces. The smoothness assumption is not enough to capture the complex statistics of natural 3D surfaces. In Chapter 5, I describe our approach for modeling the priors of surface shape. Finally, Saxena's algorithm relies on some regularities specific to their range image database. Each image in their database is of an open indoor or outdoor scene, centered on the horizon. The algorithm learns a different set of parameters for each row of the output range image. Thus, the algorithm implicitly requires prior knowledge of the inclination of each pixel. This approach is well suited to the open scenes in their database, but may fail at inferring the 3D shapes of individual objects. My goal, on the other hand, is to develop a general depth inference approach that should work for images of objects as well as open scenes.

Another recent work that seeks to infer coarse 3D geometric structure from single images is Derek Hoiem's work on Automatic Pop-up [34, 33]. In [34], image regions are categorized into ground plane, sky, vertical surfaces facing left, right, or towards the camera, non-planar porous surfaces (such as foliage or wire fencing), and non-planar "solids" (e.g. people or tree trunks). The algorithm works by first segmenting the image into small regions called "super-pixels", which are then grouped into larger regions. Then, each region is assigned a label according to the properties of the image within that region, where label assignment is learned from a database of hand-labeled images. Logistic regression Adaboost using decision trees are used to learn the label-assignment likelihood function. Image cues include region intensity and color, texture statistics, region location, size, and contour shape, and line geometry statistics designed to relate to perspective cues (e.g. a line orientation histogram, and statistics of the locations of line intersections within the region). Labels are assigned using bottom-up image cues exclusively; each label is assigned independently of its neighbors, and no computational interaction is necessary between regions. Instead of complex recurrent computations, the algorithm draws its strength from a good bottom-up image segmentation. Even in [33], where the only categories are ground plane, vertical, and sky, a reasonable coarse scene geometry is often produced in certain scenes. Such successes illustrate the power of good localization of occlusion contours.

Chapter 3

Methods of Statistical Inference

As discussed in the introduction, a statistical approach to the depth inference problem promises to offer several advantages over deterministic approaches. A statistical approach should improve the generality and robustness of the algorithm, allowing reasonable shape estimates to be inferred in a wider class of images. A statistical approach can expoit stochastic depth cues which were previously inaccessable or whose deterministic forms were too mathematically cumbersome to manage. Statistical methods allow us to estimate measures of confidence of various aspects of the reconstructed 3D surface, rather than a single point-estimate alone. Finally, statistical approaches can facilitate the combination of multiple cues, allowing conflicting evidence from multiple sources to be resolved according to the confidence of each source.

In order for a statistical approach to depth inference to be successful, we must have a method for statistical inference that is capable of finding likely values within a joint probability distribution of shape and appearance that is both very large and very complex. Most existing methods of statistical inference are not equiped to handle the size and complexity of the problems of depth inference discussed later in this thesis. In this section, I will begin by briefly reviewing some previous relevant methods for statistical inference, and the advantages and drawbacks of each. In section 3.3, I will present a more in-depth look at *belief propagation*, a highly promising method of statistical inference that has lead to great progress in a number of applications. Unfortunately, for problems with the level of complexity we wish to address, belief propagation is intractible, requiring years of computation to solve even highly restricted depth estimation problems such as shape-from-shading. In chapter 4, I will introduce a technique that reduces the computational com-

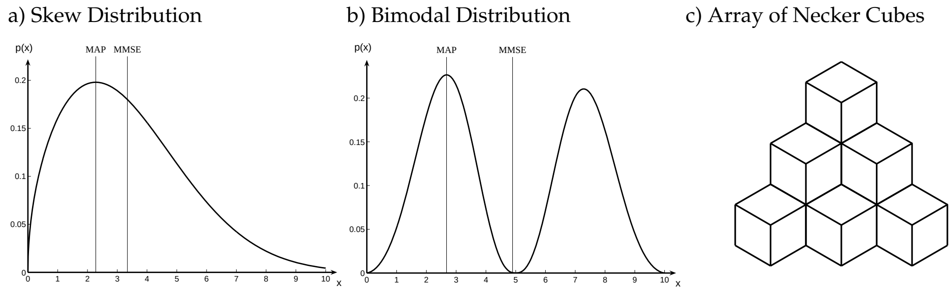


Figure 3.1: The relative advantages of MAP and MMSE point estimates depend on the distribution in question (see text for discussion). Subfigure **c)** shows an ambiguous 3D shape. A MAP estimator will choose one of the two likely interpretations, whereas the MMSE estimator will average the two together, resulting in a flat surface estimate.

plexity of belief propagation from exponential to linear in the clique size of the underlying graphical model. These advancements will allow us to efficiently solve inference problems that were previously unavailable to belief propagation.

3.1 Defining the Problem: A Statistical Approach to Depth Inference

I begin by describing the problem from the perspective of statistical inference. Consider the space of all possible of images I. This space is extremely large, and most of the possible images are meaningless. Such meaningless images are less likely to be observed in nature than images depicting real objects. We can imagine a probability distribution $p(I)$ that represents the likelihood of encountering each image in nature. Similarly, we can imagine a joint probability distribution $p(I, Z)$ over all pairs of images I and range images Z. Ideally, to infer likely range images Z given an image I, we would like to model the posterior distribution $p(Z|I) = p(I, Z)/p(I)$. We could then compute the optimal 3D scene estimate for a particular image by using one of a number of loss functions, such as the mode of posterior (MAP, or maximum a posteriori):

$$Z_{MAP} = \underset{Z}{\operatorname{argmax}}\, p(Z|I) \tag{3.1}$$

or the mean of the posterior (MMSE, or minimum mean-squared error):

$$Z_{MMSE} = E[Z|I] = \int Zp(Z|I)dZ \tag{3.2}$$

In some applications, the MMSE estimator is regarded as superior to the MAP estimator because the MAP estimator is insensitive to the degree of uncertainty around the mode in the distribution. For example, in a single-variate skewed distribution like the one in figure 3.1**a**, it often makes sense to choose a point estimate that is to the right of the mode. The MMSE estimator reports the center of mass of this distribution. However, this logic depends on embedding the state space of possible configurations in a metric space where weighted averages over different configurations are meaningful. For ambiguous images, the posterior distribution is often highly bimodal (as in figure 3.1**b**). One common example of a scene with ambiguous 3D interpretation is the Necker cube (figure 3.1**c**). This cube can be perceived as protruding out of the page, or receding into it. The MMSE estimator averages over all likely 3D shapes, and thus reports a flat surface - a highly unlikely result. The MAP estimate, on the other hand, must select one of the two likely interpretations. Regardless of what point estimator we use, it would be advantageous if some estimate of the uncertainty of the distribution were also computed, in addition to a point estimate. Specifically, a MMSE estimate with a set of marginals is more useful than a MMSE estimate alone.

3.2 Methods of Statistical Inference

The problem of statistical inference is central to artificial intelligence and to computation in general. Unfortunately, in the general case, finding the MAP or MMSE point estimate of a distribution is NP-Hard [80]. Thus, for large, complex distributions like $p(Z|I)$, approximate methods must be used to estimate the MAP or MMSE points of a distribution.

One simple approach is to use a gradient descent algorithm on the posterior distribution to find the MAP estimate. The problem is that gradient descent can easily become stuck in local minima. This is a serious problem for all but the most simple posterior distributions. Stochastic relaxation is a similar technique where, at each iteration, the gradient is followed with some probability proportional to temperature, T. Otherwise, some random direction is followed. The advantage of stochastic relaxation is that, if the temperature is decreased slowly enough, you are guaranteed to find the global maximum. Unfortunately, for a complex distribution this approach can be prohibitively slow. Also, it can be difficult to determine how to control the temperature. As mentioned in chapter 1, gradient descent and related methods have been tried extensively for solving the problem of shape from shading, but copious local minima appear to make this approach ineffective [109].

A related approach is Markov chain Monte Carlo (MCMC) sampling. In this family of algorithms, we seek to approximate the posterior distribution by generating a set of samples from this distribution. Sampling can be used to compute a MAP estimate by simply selecting the sample with the highest probability according to the model probability distribution. MMSE and other estimators can also be approximated from a sample list. Unfortunately, MCMC sampling can also be prohibitively slow, especially in very high dimensional problems like 3D shape inference. What's more, it is often very difficult to determine if the algorithm has converged, or if some important portion of the state space has not yet been explored by the stochastic sampling algorithm.

One key insight that has been greatly helpful for statistical inference is to exploit local structure within a probability distribution. Specifically, many probability distributions can be *factorized*, or represented as a product of *potential functions*, each of which ranges over only a small subset of variables of the problem space $\vec{X}$:

$$p(\vec{X}) = \prod \phi_i(\vec{x_i}) \qquad \vec{x_i} \subset \vec{X} \tag{3.3}$$

Graph cuts are one popular method for MAP estimation of factorized distributions which have been successful in a variety of vision applications, including stereo [45, 77] and photometric stereo [101]. However, for this method to work, the potential functions in equation 3.3 must meet a set of constraints [46]. Specifically, each potential function must be *regular*. For potential functions of two variables, this means that for any three variable states x_1, x_2, and α, we must have

$$\phi(x_1, x_2) + \phi(\alpha, \alpha) \geq \phi(\alpha, x_2) + \phi(x_1, \alpha) \tag{3.4}$$

Loosely speaking, this means that potential functions must not discourage variables from being identical. Potential functions of four or more variables may have additional constraints, as there is as of yet no known general method for constructing a graph for such potential functions. As I will show in later sections, these constraints will interfere with the construction of a strong shape prior (i.e. priors that describe more than just surface smoothness), and also for the incorporation of shading cues.

3.3 Belief Propagation

Belief propagation is a method of computing the single-variate marginals for each variable in a factor graph. The original formulation of belief propagation was designed to work in factor graphs that have the form of a tree (they contain no loops). In tree-strutured factor graphs, belief propagation is non-iterative and exact: the marginals computed by this method are equivalent to those computed using brute force:

$$p(x_i) = \sum_{X \setminus x_i} p(\vec{x})$$

Later, a variant of belief propagation known as *loopy belief propagation* was developed to apply to arbitrary factor graphs. Empirical success of loopy belief propagation was demonstrated for a variety of applications, such as decoding turbo-codes [48], image super-resolution [23], stereo [82] and photometric stereo [84].

Loopy belief propagation works by iteratively passing vector-valued messages along each edge of the factor graph according to the equations:

$$m^t_{i \to f}(x_i) = \prod_{g \in \mathcal{N}(i) \setminus f} m^{t-1}_{g \to i}(x_i) \tag{3.5}$$

$$m^t_{f \to i}(x_i) = \sum_{\vec{x}_{\mathcal{N}(f) \setminus i}} \left(\phi_f \left(\vec{x}_{\mathcal{N}(f)} \right) \prod_{j \in \mathcal{N}(f) \setminus i} m^t_{j \to f}(x_j) \right) \tag{3.6}$$

$$b^t_i(x_i) \propto \prod_{g \in \mathcal{N}(i)} m^t_{g \to i}(x_i) \tag{3.7}$$

where f and g are factor nodes, i and j are variable nodes, and $\mathcal{N}(i)$ is the set of neighbors of node i [29]. Here, $b_i(x_i)$ is the estimated marginal of variable i, meaning that

$$b(x_i) \approx p(x_i) = \sum_{X \setminus x_i} p(\vec{x}) \tag{3.8}$$

The expected value of $\vec{X}$, or equivalently, the minimum mean-squared error (MMSE) point estimate, can be computed by finding the mean of each marginal.

The above formulation of belief propagation is also known as *sum-product* belief propagation. One other variant of belief propagation is *max-product* belief propagation. The goal of max-product belief propagation is to compute the "maximals" or "max-marginals" of the distribution:

$$\hat{b}(x_i) \approx \max_{x_j \in X \setminus x_i} P(X) \tag{3.9}$$

Max-product belief propagation proceeds very similarly to sum-product belief propagation, except that the summands in equation 3.6 are replaced by maximums:

$$\hat{m}^t_{i \to f}(x_i) = \prod_{g \in \mathcal{N}(i) \setminus f} \hat{m}^{t-1}_{g \to i}(x_i) \tag{3.10}$$

$$\hat{m}^t_{f \to i}(x_i) = \max_{\vec{x}_{\mathcal{N}(f) \setminus i}} \left(\phi_f \left(\vec{x}_{\mathcal{N}(f)} \right) \prod_{j \in \mathcal{N}(f) \setminus i} \hat{m}^t_{j \to f}(x_j) \right) \tag{3.11}$$

$$\hat{b}^t_i(x_i) \propto \prod_{g \in \mathcal{N}(i)} \hat{m}^t_{g \to i}(x_i) \tag{3.12}$$

Once the maximals $\hat{b}$ are estimated, the MAP point estimate can be approximated by choosing the value of x_i that maximizes its maximal $\hat{b}_i(x_i)$.

As we mentioned above, both forms of belief propagation give exact results in factor graphs without loops. When belief propagation was first applied to networks with loops, good results were often achieved [48, 62] without any theoretical justification. Since that time, theoretical works have begun to shed some light on which graphical models loopy belief propagation is likely to work well for, and what the limits on its performance are. It was shown that, for jointly Gaussian MRFs, if the sum-product algorithm converges, the means of the computed posterior marginals (and thus the MMSE estimate) are correct [94], even though the variances of those distributions are often wrong. In the same paper, it was also shown that if the max-product algorithm converges, the resulting MAP estimate will be a local maximum of the true posterior distribution (even for non-Gaussian distributions). A subsequent paper [95] improved this result by showing that the computed MAP estimate must have a greater posterior probability than any estimate that can be computed by modifying the values of the MAP estimate in any region of nodes, provided that region contain no loops.

Later, it was discovered that when the Sum-Product algorithm converges (in pairwise-connected MRFs), the resulting marginals minimize the Bethe free energy, which can be thought of as an approximate measure of the distance between a multivariate probability distribution and a set of marginals [103]. Let $p(X)$ be the actual joint probability distribution, and let $\{b_{ij}(x_i, x_j), b_i(x_i)\}$ be the set of pairwise and single-variate marginals designed to approximate $p(X)$ (typically referred to as "belief").

In the case that all factors ϕ_{ij} of equation 3.3 are bivariate (if the distribution is a pairwise-connected MRF), the Bethe free energy is given by:

$$\begin{aligned} D_{bethe}(\{b_{ij}, b_i\}||p) =& \sum_{ij} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln \left(\frac{b_{ij}(x_i, x_j)}{\phi_{ij}(x_i, x_j)} \right) \\ & - \sum_i (q_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \end{aligned} \tag{3.13}$$

where q_i is the number of neighbors of node i. The Bethe free energy generalizes naturally to arbitrary factor graphs (see [105] for more detail). Technically, b_{ij} and b_i are often referred to as "pseudo-marginals" instead of marginals, because there is no guarantee that a joint distribution exists which has such marginals. However, the set of pseudo-marginals is constrained by normalization and marginalization constraints:

$$\sum_{x_i} b_{ij}(x_i, x_j) = b_j(x_j) \tag{3.14}$$

$$\sum_{x_i} b_i(x_i) = 1 \tag{3.15}$$

The Bethe free energy is an approximation of the Gibbs free energy, which for probability distributions as defined in equation 3.3 is equal to the Kullback-Leibler divergence

$$D_{KL}(b||p) = \sum_{x \in X} b(x) \ln \frac{b(x)}{p(x)} \tag{3.16}$$

where $b(X)$ is the approximated joint probability distribution. Intuitively, the first term of equation 3.13 sums up the KL divergence of each pairwise marginal, and the second term compensates for over-counting the divergence of the single-variate marginals.

The connection between Bethe free energy and loopy belief propagation was an important discovery, because it provided a theoretical justification for the application of belief propagation on networks with loops. Secondly, it provided a way to design improvements of loopy belief propagation by using more accurate approximations of distance between a distribution and a set of marginals, such as the more sophisticated Kikuchi energies [103, 104, 105]. Finally, the discovery of the connection between loopy belief propagation and Bethe free energy allowed more robust algorithms to be designed, which guarantee convergence on a minima of Bethe free energy by seeking to minimize the quantity directly [107, 29]. One method of convergent belief propagation is discussed in greater length in section 3.4.

The principal computational expense of loopy belief propagation is the multidimensional summation (in the case of MMSE) or maximization (in the case of MAP) in messages from factor nodes to variable nodes (equation 3.6 or 3.11). Suppose factor node f has N adjacent variable nodes (including i). To compute $m^{t+1}_{f\to i}(x_i)$, for each possible value of x_i, we must sum over a $N-1$ dimensional array for each value of x_i. This has a cost of $\mathcal{O}(M^N)$, where M is the number of possible values of each variable. A similar computation must be computed for each edge at each factor node, for a total computational cost of $\mathcal{O}(FNM^N)$ for each iteration, where F is the total number of factor nodes. For belief propagation to be tractable, M^N must remain low. Because we are attempting to infer 3D shape, which is comprised of many continuously-valued random variables, there is a limit as to how low M can be. Because of this, N must be kept quite low. Most computer vision applications of loopy belief propagation that infer non-binary valued scene properties use a value of N no higher than 2 [23, 82, 84]. In Chapter 4, I introduce a computational shortcut that reduces the complexity of computing belief propagation messages from $\mathcal{O}(M^N)$ to $\mathcal{O}(NM^2)$, or from exponential to linear in the size of each clique. This technique creates an opportunity to apply a sophisticated and successful inference technique to complex statistical problems where belief propagation was previously intractable.

3.4 Convergent Loopy Belief Propagation

One of the biggest shortcomings of loopy belief propagation is that it is not guaranteed to converge. Convergence becomes increasingly unlikely when the factor graph contains many tight loops, or when potential functions are "high energy," or nearly deterministic [28]. The depth inference problems that we will be considering later exhibit both of these problems, and empirically, they often fail to converge using standard belief propagation, even using different dampening, scheduling, or reweighting techniques.

Fortunately, it was recently discovered that when standard sum-product loopy belief propagation converges, the resulting marginals minimize a quantity from statistical physics known as the Bethe free energy [103]. This has lead to the development of belief propagation algorithms that minimize the Bethe free energy directly [106, 29], and do so while ensuring convergence.

In the examples presented here, we use the algorithm described in [29], which modifies equations 3.5 and 3.6 by:

$$m^t_{i\to f}(x_i) = m^{t-1}_{f\to i}(x_i)^{\frac{1-n_i}{n_i}} \prod_{g\in\mathcal{N}(i)\setminus f} m^{t-1}_{g\to i}(x_i)^{\frac{1}{n_i}} \tag{3.17}$$

$$m^t_{f\to i}(x_i) = \int_{\vec{x}_{\mathcal{N}(f)\setminus i}} \tilde{\phi}_f\left(\vec{x}_{\mathcal{N}(f)}\right) \prod_{j\in\mathcal{N}(f)\setminus i} m^t_{j\to f}(x_j)\, d\vec{x} \tag{3.18}$$

$$b^t_i(x_i) \propto \prod_{g\in\mathcal{N}(i)} m^t_{g\to i}(x_i)^{\frac{1}{n_i}} \tag{3.19}$$

where $n_i = |\mathcal{N}(i)|$, the number of neighbors of variable node i. Initially, $\tilde{\phi}_f$ is set to equal ϕ_f. Each time the estimated beliefs in equation 3.19 converge, $\tilde{\phi}_f$ is updated according to

$$\tilde{\phi}_f(\vec{x}_{\mathcal{N}(f)}) = \phi_f(\vec{x}_{\mathcal{N}(f)}) \prod_{j\in\mathcal{N}(f)} b^\tau_j(x_j)^{\frac{n_j-1}{n_j}} \tag{3.20}$$

where $b^\tau_j(x_j)$ is the belief at variable node j the last time the algorithm converged. The algorithm continues until $b^\tau_j(x_j)$ itself converges.

Not only does this approach guarantee convergence, but we have found that the results are often superior to standard LBP when standard LBP does converge.

One drawback to Heskes' convergent algorithm is that it is not compatible with max-product belief propagation. However, when maximum a-posteriori point estimates are desired, we can achieve them using the approach proposed by Yuille [106], which introduces a temperature T, and replaces the energy function of equation 7.2 with $\prod \phi_i(\vec{x_i})^{\frac{1}{T}}$. As the algorithm converges, T is reduced. As T approaches zero, the computed marginals will approximate the "maximals" of max-product belief propagation.

Another method for improving the performance and convergence properties of the original belief propagation equation is to use *tree reweighting* methods [92, 44, 87]. Tree-reweighted extensions to belief propagation apply to both sum-product belief propagation [92, 87] and max-product belief propagation [44].

Chapter 4

Efficient Belief Propagation

As discussed in Chapter 3, belief propagation is a powerful method of statistical inference that has contributed to great progress for a wide variety of applications [48, 23, 82, 84, 51, 65, 93]. Unfortunately, the computational complexity of belief propagation grows very quickly with the complexity of the underlying factor graph, and it has previously been feasible only for very simple statistical models. Specifically, computing belief propagation messages is exponential in the size of the largest graph clique. This means that for problems with many labels or real-valued variables, belief propagation methods have historically been limited to graphical models with only pairwise interactions between variables. Unfortunately, pairwise connected models are often insufficient to capture the full complexity of the joint distribution of the problem. This is especially true in computer vision, where the rich and complex statistical structure of natural images cannot be captured by pairwise connected Markov Random Fields [38].

In this section, I introduce a series of methods that reduce the computational complexity of LBP, and make LBP feasible for wider classes of statistical inference problems. In section 4.1, I propose a computational shortcut that, for a wide class of potential functions, reduces the complexity of belief propagation from exponential in clique size, to linear in clique size. This technique allows the highly successful belief propagation algorithm to be applied to rich, complex statistical inference problems that will be instrumental to solving problems related to depth-inference discussed in chapters 5 and 6. This method is exact, and achieves efficient belief propagation without sacrificing accuracy. In sections 4.2 and 4.3, I demonstrate how this technique can be applied to wider subclasses of potential functions. In section 4.4, I examine different methods of representing the messages of belief propagation, which can effect the efficiency and performance of the algorithm. Finally, in section 4.5, I propose a technique that allows belief propagation messages to be computed efficiently for arbitrary potential functions. This most general method is approximate, but can approximate belief propagation messages arbitrarily well given sufficient resources.

4.1 Efficient Belief Propagation Using Linear Constraint Nodes

In this section, I introduce a new technique to compute belief propagation messages in time linear with respect to clique size that works for a large class of potential functions without resorting to approximation. This technique allows us to apply powerful belief propagation methods to complex, real-valued statistical inference problems that were previously intractible. For continuous, real-valued random variables, the equations for belief propagation are

$$m^t_{i\to f}(x_i) = \prod_{g\in\mathcal{N}(i)\backslash f} m^{t-1}_{g\to i}(x_i) \tag{4.1}$$

$$m^t_{f\to i}(x_i) = \int_{\vec{x}_{\mathcal{N}(f)\backslash i}} \phi_f\left(\vec{x}_{\mathcal{N}(f)}\right) \prod_{j\in\mathcal{N}(f)\backslash i} m^t_{j\to f}(x_j)\, d\vec{x} \tag{4.2}$$

$$b^t_i(x_i) \propto \prod_{g\in\mathcal{N}(i)} m^t_{g\to i}(x_i) \tag{4.3}$$

This formulation is the same as in equations 3.5 through 3.7, except the summations are replaced with integrands.

In practice, the integrals performed in belief propagation equation 4.2 typically cannot be computed or represented analytically. In these cases, the beliefs $b_i(x_i)$ and messages $m_{i\to f}(x_i)$ are often approximated by discrete histograms. When messages are represented by histograms, the integrand of equation 4.2 is replaced by a summand. Thus, the equations for belief propagation become:

$$m^t_{i\to f}(x_i) = \prod_{g\in\mathcal{N}(i)\backslash f} m^{t-1}_{g\to i}(x_i) \tag{4.4}$$

$$m^t_{f\to i}(x_i) = \sum_{\vec{x}_{\mathcal{N}(f)\backslash i}} \left(\phi_f\left(\vec{x}_{\mathcal{N}(f)}\right) \prod_{j\in\mathcal{N}(f)\backslash i} m^t_{j\to f}(x_j) \right) \tag{4.5}$$

$$b^t_i(x_i) \propto \prod_{g\in\mathcal{N}(i)} m^t_{g\to i}(x_i) \tag{4.6}$$

and the algorithm proceeds as before. In the next several sections, we will assume that messages are represented using discrete histograms. We will continue to write the belief propagation equations in continuous form, so that the error of discretization can be postponed for as long as possible. Later, in section 4.4, I will discuss alternate methods of message representation, their implications for belief propagation inference in networks with higher order cliques, and ways of minimizing discretization error.

In this section, I will continue to use the simpler, original formulation of loopy belief propagation, rather than the convergent variants discussed in section 3.4. However, the methods we introduce in this section and in following sections all generalize easily to these convergent variants. Later, in section 4.4, I will discuss in more detail the implications of convergent variants of belief propagation on the efficient computation and representation of messages.

4.1.1 Linear Constraint Nodes

Consider potential functions of the form

$$\phi(\vec{x}) = g(\vec{x}\cdot\vec{v}) \tag{4.7}$$

where $\vec{x}$ and $\vec{v}$ are vectors of length N. Factor nodes of this form will be referred to as Linear Constraint Nodes (LCNs). Normally, computing messages from such factor nodes takes $\mathcal{O}(M^N)$ time. Here, we show that, using a change of variables, this computation can be done in $\mathcal{O}(NM^2)$ time. For notational simplicity, we illustrate this using $N = 4$, although the method extends easily to arbitrary N. For shorthand, let $M_i \equiv m_{f\to i}$ and $m_i \equiv m_{i\to f}$

Then we have:

$$\begin{aligned} M_1(x_1) = & \iiint g(v_1x_1 + v_2x_2 + v_3x_3 + v_4x_4) \\ & \qquad m_2(x_2)m_3(x_3)m_4(x_4)\, dx_2\, dx_3\, dx_4 \qquad (4.8) \\ = & \iiint J\, g(v_1x_1 + y_2)\, m_2(\frac{y_2 - y_3}{v_2}) \\ & \qquad m_3(\frac{y_3 - y_4}{v_3})\, m_4(\frac{y_4}{v_4})\, dy_2\, dy_3\, dy_4 \qquad (4.9) \\ \propto & \int g(v_1x_1 + y_2)\Bigg(\int m_2(\frac{y_2 - y_3}{v_2}) \\ & \Bigg(\int m_3(\frac{y_3 - y_4}{v_3}) m_4(\frac{y_4}{v_4}) dy_4 \Bigg) dy_3 \Bigg) dy_2 \qquad (4.10) \end{aligned}$$

where J is the (constant) Jacobian corresponding to the change of variables. Since belief propagation messages are only defined up to a constant for most variations of LBP, the Jacobian can be safely ignored in this case. Here we have used the change of variables:

$$y_4 = v_4x_4 \qquad (4.11)$$

$$y_3 = v_3x_3 + y_4 \qquad (4.12)$$

$$y_2 = v_2x_2 + y_3 \qquad (4.13)$$

$$J = 1/(v_2v_3v_4) \qquad (4.14)$$

This allows us to perform each integrand one at a time. Since each of the $N-1$ integrands depend only on two variables, each can be computed in $\mathcal{O}(M^2)$ time. In section 4.4.3, we provide more technical details on how to compute these integrals for histogram-based message representations, and show that the method of computing messages described here not only results in a significant computational speed-up, but also lowers the discretization error.

The transformation of variables used above works for any vector $\vec{v}$. However, there are many possible transformations. Clever choice of the transformation of variables may allow one to reuse intermediate computations during the computation of other messages, or to embed additional nonlinear potential functions of pairs of variables y_i and y_{i+1} at no extra computational cost. The choice of transformation of variables is discussed further in section 4.3.

If $v_i = \pm 1$ for all i, and messages are represented as uniform-width histograms, then each integrand in equation 4.10 can be reduced to a $\mathcal{O}(M \log M)$ computation using discrete Fourier transforms as in [21]. Although we describe our approach for sum-product belief propagation, the same approach is valid for max-product belief propagation. For max-product belief propagation, each maximal in equation 4.10 can be closely approximated in linear time using the distance transform methods described in [21].

4.1.2 Linear Constraint Nodes and Projection Pursuit Density Estimation Methods

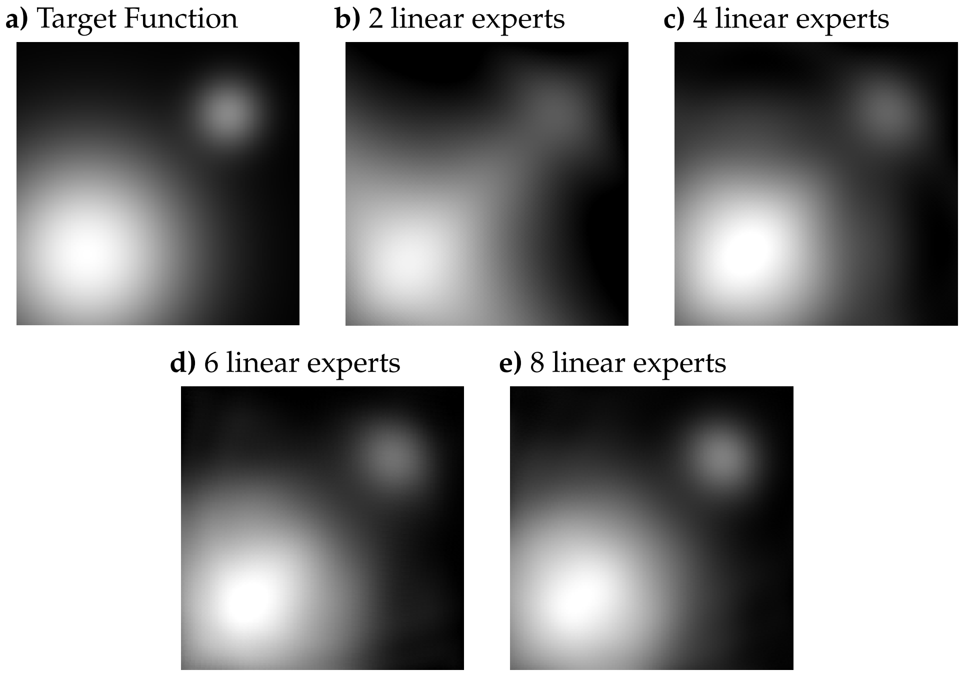
Systems of linear constraint nodes, of the form

$$P(\vec{x}) \approx \tilde{P}(\vec{x}) = \prod_{k=1}^{K} g_k(\vec{x} \cdot \vec{v}_k) \tag{4.15}$$

have been very successful in approximating multivariate, continuous probability distributions $P(\vec{x})$. Projection pursuit density estimation [24], Minimax Entropy and FRAME [112, 113], Products of Experts [30], and Fields of Experts [73] all work by approximating distributions $P(\vec{x})$ as products of linear constraint nodes (as in equation 4.15). Previously, performing inference over these graphical models typically required using gradient descent or related methods. These approaches often struggled with local maxima. In Chapter 5, I will show how the shortcut introduced in section 4.1.1 allows us to perform inference in Fields of Experts using belief propagation. Our results significantly outperform gradient descent based methods of optimization.

Products of linear potential functions have several attractive features that have lead to their success. Their factorized nature simplifies the problem of learning parameters, and several powerful methods for learning parameters g_k and $\vec{v}_k$ have been developed [30, 96, 85, 11]. Additionally, systems of linear potential functions as in equation 4.15 are members of the exponential family of probability density models [112]. One consequence of this is that when the potential functions g_k are learned so as to minimize the KL-divergence between $P(\vec{x})$ and $\tilde{P}(\vec{x})$

$$D_{KL}[P(\vec{x})||\tilde{P}(\vec{x})] = \int P(\vec{x}) \log \frac{P(\vec{x})}{\tilde{P}(\vec{x})} d\vec{x} \tag{4.16}$$



a) Target Function

b) 2 linear experts

c) 4 linear experts

d) 6 linear experts

e) 8 linear experts

Figure 4.1: Illustrating how products of linear potential functions can approximate arbitrary functions. **a)** The target potential function to be approximated: a two-dimensional mixture of Gaussians. Subfigures **b)** through **e)** show the target function approximated with an increasing number of linear potential functions. Vectors $\vec{v}_k$ were chosen to manually to be evenly spaced.

(or equivalently, learned so as to maximize the log likelihood of the training data), the single-variate marginals of $\tilde{P}(\vec{x})$ projected onto each vector v_k will match those of the target distribution $P(\vec{x})$:

$$\int P(\vec{x})\delta(\vec{x}\cdot\vec{v_k}-\rho)d\vec{x} = \int \tilde{P}(\vec{x})\delta(\vec{x}\cdot\vec{v}-\rho)d\vec{x} \qquad \forall\rho,k \tag{4.17}$$

Furthermore, of all probability distributions that share this property (those that satisfy equation 4.17), $\tilde{P}(\vec{x})$ will achieve the maximal possible entropy [112]. Intuitively, this suggests that $\tilde{P}$ makes as few assumptions as possible regarding features $\vec{v}'$ that the model was not trained on.

Finally, we point out that, given enough linear potential functions, the product of those potential functions can approximate any probability distribution or desired nonlinear potential function arbitrarily well. Suppose we allow K to approach infinity. Then equation 4.15 becomes

$$\log \tilde{P}(\vec{x}) = \int_{|v|=1} g_{\vec{v}}(\vec{x} \cdot \vec{v}) d\vec{v} \tag{4.18}$$

Now consider the Radon transform

$$\mathcal{R}[f(\vec{x})](\rho, \vec{v}) = \int f(\vec{x}) \delta(\vec{x} \cdot \vec{v} - \rho) d\vec{x} \tag{4.19}$$

where $\vec{v}$ is constrained to be of unit norm. The adjoint of the Radon transform [18] has the form

$$\mathcal{R}^{\dagger}[\psi(\rho, \vec{v})](\vec{x}) = \int_{|v|=1} \psi(\vec{x} \cdot \vec{v}, \vec{v})\, d\vec{v} \tag{4.20}$$

The Radon transform is invertible [18], and since the adjoint of an invertible function is itself invertible, equation 4.20 is also invertible. This means that we can always choose our potential functions $g_{\vec{v}}(\rho)$ in such a way that $\tilde{P}(\vec{x}) = P(\vec{x})$ exactly. Specifically, choosing $g_{\vec{v}}(\rho) = \mathcal{R}^{\dagger -1}[\log P(\vec{x})]$ results in a perfect reproduction of the target probability distribution $P(\vec{x})$. In practice, large values of K are often impractical. However, in our experience, all but the most pathological probability density functions $P(\vec{x})$ can be approximated well with only a small number of linear potential functions. In figure 4.1, we illustrate how a product of several linear potential functions can be used to approximate an arbitrary function.

4.1.3 Hard Linear Constraint Nodes

A subclass of linear constraint nodes that is especially useful is the *hard linear constraint node*. Hard linear constraint nodes have the form:

$$\phi(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \cdot \vec{v} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

or equivalently, hard linear constraint nodes have a nonlinearity g that is a delta function. We refer to linear constraint nodes that are *not* of this form as *soft linear constraint nodes.*

Hard linear constraint nodes are useful because they enforce linear dependencies among the variable nodes in a graphical model. For example, a hard linear constraint node may enforce that variables a, b, and c obey the relationship $a + b = c$. This ability to enforce linear dependencies means that hard linear constraint nodes allow us to utilize overcomplete representations of the problem space $\vec{X}$. Specifically, a factor graph that uses an overcomplete representation is one that has more variable nodes than there are degrees of freedom in the underlying probability distribution. When the representation of $\vec{X}$ is overcomplete, then there must be linear dependencies among the variables of $\vec{X}$ of the form $\vec{x} \cdot \vec{v} = 0$. These dependencies must be enforced to prevent computing estimates that are internally inconsistent. Using standard belief propagation (equation 4.5), enforcing such constraints would be intractable. Using the methods in equation 4.10, these constraints can be efficiently enforced using a set of hard linear constraint nodes.

For any computational problem, finding the best way to represent the problem state space is crucial; some problems can be solved much more easily given the right representation. A single complete representation forces us to decide on only one representation, whereas overcomplete representations allow us to retain the benefits of multiple complete representations. One example of the use of overcomplete representations is multi-scale approaches in computer vision, which have been very successful in several domains. Another example can be found in the primate visual cortex, which is overcomplete by a factor of at least 200:1 relative to retinal input.

In figure 4.2, we demonstrate how hard linear constraint nodes may be used to exploit multiple-resolution techniques with belief propagation. Multiple-resolution methods, and similar approaches such as wavelet-domain processing and image-pyramid techniques, are all highly successful in computer vision, and have contributed to algorithms for image denoising [66], shape-from-stereo [98], motion [6], texture classification [59], region classification [49], and segmentation [9]. Previous statistical inference approaches that exploited multiple-resolution methods were limited to simple Gaussian models or gradient descent optimization methods. The use of hard linear constraint nodes makes multiple-resolution representations available to belief propagation techniques.

Another example of an overcomplete representation often used in vision is surface normal maps used to represent 3D surface shape. Such maps, or "needle maps," are typically represented using two values per pixel: $p = \frac{\partial z}{\partial x}$ and $q = \frac{\partial z}{\partial y}$. For any real surface $z(x, y)$, its gradient field must satisfy the

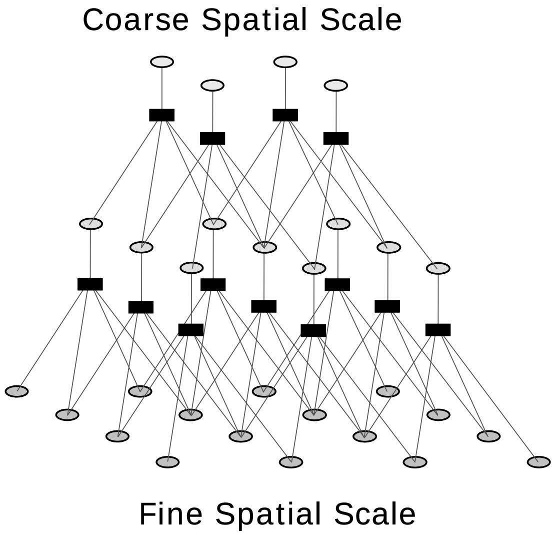


Figure 4.2: A factor graph that demonstrates the use of multiple resolution inference for belief propagation. Each circle represents a variable at one of three spatial scales, and each black square represents a hard linear constraint factor node. Here, each hard linear constraint node enforces that its upper neighbor is the block average of the pixels in the next finer spatial scale. Wavelet and Laplacian image pyramids are also possible. The methods of section 4.1.1 reduce the number of operations required to compute belief propagation messages in this network from $\mathcal{O}(M^5)$ to $\mathcal{O}(M^2)$.

zero curl requirement, or equivalently,

$$\frac{\partial}{\partial y}\left(\frac{\partial z}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{\partial z}{\partial y}\right) \tag{4.22}$$

$$\frac{\partial}{\partial y}p = \frac{\partial}{\partial x}q \tag{4.23}$$

In the computer vision literature, this equality also referred to as the integrability constraint, which ensures that a surface's normal map must integrate to a valid surface z. When p and q do not satisfy this relationship, there is no surface $z(x, y)$ that is consistent with p and q. In discrete form, the integrability constraint is equivalent to

$$p(x, y) - q(x, y) + q(x\ +\ 1, y) - p(x, y\ +\ 1) = 0 \tag{4.24}$$

where

$$p(x,y) = z(x\ +\ 1,y) - z(x,y) \tag{4.25}$$
$$q(x,y) = z(x,y\ +\ 1) - z(x,y) \tag{4.26}$$

The integrability constraint can be enforced efficiently using a hard linear constraint node of four variables. For many problems of 3D shape inference, representing shape using a surface normal map can be a great advantage. Consider the classic problem of shape-from-shading, where the image intensity at each point restricts the surface normal to lie along some one-dimensional manifold, according to the Lambertian equation:

$$i(x,y) = \max(0, \frac{1 + pL_p + qL_q}{\sqrt{1+p^2+q^2}\sqrt{1+L_p^2+L_q^2}}) \tag{4.27}$$

where L_p and L_q specify the lighting direction. This relationship between p and q could be implemented as a pairwise clique in an overcomplete factor graph with the integrability constraint enforced using hard linear constraint nodes of clique-size four (as done in Chapter 6). Alternatively, the Lambertian relationship could be enforced using cliques of size three in a complete factor graph whose variable nodes represent depth at each pixel:

$$i(x,y) = \max(0, \frac{s(x,y)}{\sqrt{1+L_p^2+L_q^2}}) \tag{4.28}$$
$$s(x,y) = \frac{1 + (z_{x+1,y} - z_{x,y})L_p + (z_{x,y+1} - z_{x,y})L_q}{\sqrt{1 + (z_{x+1,y} - z_{x,y})^2 + (z_{x,y+1} - z_{x,y})^2}} \tag{4.29}$$

However, note that because absolute depth is completely ambiguous in shape-from-shading, the computed marginals of z should be expected to be highly uniform over a large range of depths. Even if an absolute depth is arbitrarily chosen at one node, belief propagation is then charged with the task of propagating this value to all nodes in the image. Since uncertainty compounds over space, this measure would be ineffective outside of a small radius. Thus, using an overcomplete representation in this case is essential.

Another useful application of hard linear constraint nodes is the ability to aggregate over a set of local data to compute global features, such as by summing over several variable nodes. For example, in [58], the authors seek to infer the location and activity of a person from a stream of several days worth of GPS coordinates. In order to place a prior over the number of times a given activity occurs in a single day, variable nodes representing individual activities must be summed over. In [58], techniques similar to hard linear constraint nodes are used to perform belief propagation efficiently, where a tree of variable nodes is constructed, each node summing over two children. The methods of this paper show that such a tree structure can be replaced by a single hard linear constraint factor node, by setting $\vec{v}$ in equation 4.7 to $[-1, 1, 1, \ldots, 1]$. This would reduce the memory requirements by half (without increasing the number of operations), and, for convergent variants of belief propagation (discussed in section 3.4), would reduce the number of iterations of belief propagation required. The results of this paper also show how belief propagation can be made efficient for a much larger class of potential functions, including other examples that can be used to aggregate data across many variable nodes. For example, section 4.3, we show how variable nodes that extract the maximum value from a stream of local variable nodes can also be made efficient.

4.2 Nonlinear Constraint Nodes

We now extend our method to include potential functions of the form

$$\phi(\vec{x}) = g(g_1(x_1) + \cdots + g_N(x_N)) \tag{4.30}$$

For the sake of brevity, we consider the case where $N = 3$, although the same method works for cliques of arbitrary size. If g_i is invertible for all i, then we can apply a change of variables to equation 4.2 to get:

$$\begin{aligned} M_1(x_1) =& \iint g(g_1(x_1) + g_2(x_2) + g_3(x_3)) \\ & \qquad m_2(x_2)\, m_3(x_3)\, dx_2\, dx_3 \qquad (4.31) \\ =& \iint J(\hat{x}_2, \hat{x}_3)\, g(g_1(x_1) + \hat{x}_2 + \hat{x}_3) \\ & \qquad m_2(g_2^{-1}(\hat{x}_2))\, m_3(g_3^{-1}(\hat{x}_3))\, d\hat{x}_2\, d\hat{x}_3 \qquad (4.32) \end{aligned}$$

where we have applied the change of variables

$$\hat{x}_2 = g_2(x_2) \tag{4.33}$$

$$\hat{x}_3 = g_3(x_3) \tag{4.34}$$

$$J(\hat{x}_2, \hat{x}_3) = \left(\frac{\partial}{\partial \hat{x}_2} g_2^{-1}(\hat{x}_2)\right) \left(\frac{\partial}{\partial \hat{x}_3} g_3^{-1}(\hat{x}_3)\right) \tag{4.35}$$

The Jacobian $J(\hat{x}_2, \hat{x}_3)$ can be absorbed into the messages by defining

$$\hat{m}_i(\hat{x}_i) = m_i(g_i^{-1}(\hat{x}_i))\frac{\partial}{\partial \hat{x}_i} g_i^{-1}(\hat{x}_i) \tag{4.36}$$

and so we have

$$M_1(x_1) = \int\int g(g_1(x_1)+\hat{x}_2+\hat{x}_3)\hat{m}_2(\hat{x}_2)\hat{m}_3(\hat{x}_3)d\hat{x}_2 d\hat{x}_3 \tag{4.37}$$

We can then apply the methods of section 4.1.1 to get

$$M_1(x_1) \propto \int g(g_1(x_1)+y_2)\int \hat{m}_2(y_2-y_3)\hat{m}_3(y_3)\,dy_3\,dy_2 \tag{4.38}$$

where we have made the change of variables $y_2 = \hat{x}_2 + \hat{x}_3$ and $y_3 = \hat{x}_3$.

If g_i is not invertible, we can still apply the same technique if we integrate equation 4.2 separately for each branch of $g_i^{-1}(x_i)$. For example, if $g_i(x_i) = x_i^2$, simply integrate over the range $(-\infty, 0]$, and then over the range $(0, +\infty)$, and add the two integrals together. $g_i(x_i)$ has an inverse within both of these ranges. If the inverse of $g_i(x_i)$ has many branches, this approach may not be feasible in practice. However, nonlinearities $g_i(x_i)$ with a great many branches are not expected to come up often in real-world applications.

Using these techniques, belief propagation can be performed efficiently for a wide range of high dimensional potential functions. These include all axis-aligned generalized Gaussian distributions and Gaussian Scale Mixtures, which are popular for natural image models and denoising [66]. Since additional nonlinear potential functions of pairs of variables y_i and y_{i+1} can be embedded into equation 4.38 at no additional computational cost, many non axis-aligned Gaussians and other potential functions can also be computed efficiently using these methods.

4.3 Transformed Variable Elimination

The computational shortcuts introduced in the previous sections can be made even more general, and to apply to an even larger class of potential functions. In this section, we widen the class of potential functions that can benefit from the efficient belief propagation techniques developed so far, and at the same time, place these techniques in a broader computational framework that provides a different perspective into how these computational speed-ups are achieved, and how these methods can be tailored to suit specific applications.

For higher-order cliques, the problem of computing messages

$$m^t_{f\to i}(x_i) = \sum_{\vec{x}_{\mathcal{N}(f)\setminus i}} \phi_f\left(\vec{x}_{\mathcal{N}(f)}\right) \prod_{j\in\mathcal{N}(f)\setminus i} m^t_{j\to f}(x_j) \tag{4.39}$$

is not unlike the problem of computing a single-variate marginal

$$P_i(x_i) = \sum_{X\setminus x_i} P(X) \tag{4.40}$$

Thus, belief propagation exploits the factorization of a high-dimensional probability distribution to decompose a difficult problem (exponential in the dimensionality of X) into several easier, but similar problems (each exponential in N, the dimensionality of the clique).

When $P(X)$ can be factorized (as in equation 7.2), single variate marginals can be computed efficiently using the *Variable Elimination Algorithm* [108]. Note that this algorithm differs from belief propagation in that rather than computing *all* single-variate marginals of a distribution the elimination algorithm finds the marginal of only one variable. The variable elimination algorithm works by choosing a variable $x_j \in X \setminus x_i$, and then summing over all terms ϕ_k that depend on x_j. For example, if $P(X) = f_1(x_1, x_2, x_3)f_2(x_3, x_4)$, then eliminating the variable x_4 would proceed as:

$$\begin{aligned} P_i(x_1) &= \sum_{x_2}\sum_{x_3}\sum_{x_4} f_1(x_1, x_2, x_3)f_2(x_3, x_4) && (4.41)\\ &= \sum_{x_2}\sum_{x_3} f_1(x_1, x_2, x_3)\sum_{x_4} f_2(x_3, x_4) && (4.42)\\ &= \sum_{x_2}\sum_{x_3} f_1(x_1, x_2, x_3)g(x_3) && (4.43)\end{aligned}$$

The variable elimination process is repeated until all variables other than x_i have been eliminated. The computational complexity of the variable elimination algorithm depends on the structure of the factorization, and also on the order of elimination chosen. When the order is optimal, the complexity of the variable elimination algorithm is $\mathcal{O}(NM^{T+1})$, where M is the number of states of each variable, and T is the *treewidth* of the Markov Random Field (MRF) underlying the factorization of $P(X)$ (see [7] for a review of the treewidth of a graph). Unless the graph is very dense, $T+1$ is typically less than the number of variable nodes in the graph, and so the variable elimination algorithm represents a substantial improvement over brute force summation to compute a single-variate marginal.

If it was possible to use the variable elimination algorithm to more efficiently compute a belief propagation message (equation 4.39), then it also would have been possible to further factorize the clique potential function $\phi_f(x_{\mathcal{N}(f)})$ into a product of smaller, more efficient potential functions. Thus, we can assume that ϕ_f does not factorize, and so a direct application of the variable elimination algorithm cannot help to make belief propagation more efficient. The key insight of using linear constraint nodes is that by applying a transform $\mathcal{T}$ to the space $X \setminus x_i$ we may be factorize the transformed potential function, and so be able to apply variable elimination to an otherwise unfactorable clique.

By framing the methods of section 4.1 in this way, we can illustrate how these methods can be extended to a larger class of potential functions ϕ_f. So far, the methods of this paper has focused on finding transforms of $\phi_f(x_{\mathcal{N}(f)})$ that result in an underlying MRF in the form of a tree. A tree has a treewidth of one. Thus, once a MRF is in tree form, variable elimination can be used to compute the marginal of any node in $\mathcal{O}(NM^2)$ time. It is also possible to consider transforms $\mathcal{T}$ that transform the clique into other graphs of bounded treewidth that still represent a computational advantage over brute force summation. This allows us to improve the performance of a wider class of potential functions ϕ_f.

Let us restrict ourselves for now to linear transforms $\mathcal{T}$. Let $\mathcal{M}$ be the inverse transform matrix, so that $\mathcal{M}\vec{y} = \vec{x}$, for $\vec{x} \in X$. $\mathcal{M}$ must be an invertible matrix, and it must preserve x_i. Without loss of generality, we assume that $i = 1$, and so the top row of $\mathcal{M}$ must be $(1, 0, 0, ..., 0)$. Using transform $\mathcal{T}$, computing belief propagation messages now becomes

$$\begin{aligned} m^t_{f\to i}(x_1) &= m^t_{f\to i}(y_1) && (4.44)\\ &= J_{\mathcal{M}} \int_{Y\setminus y_1} \phi_f\left(\mathcal{M}\vec{y}\right) \prod_{j\in\mathcal{N}(f)\setminus i} m^t_{j\to f}(\mathcal{M}_{j*}\cdot\vec{y})d\vec{y} && (4.45)\end{aligned}$$

where $J_{\mathcal{M}}$ is the Jacobian of $\mathcal{M}$, and $\mathcal{M}_{j*}$ is the j^{th} row of $\mathcal{M}$. The goal of using a transform $\mathcal{T}$ is to choose a transform such that ϕ factorizes under $\mathcal{T}$:

$$\phi_f(\vec{x}) = \phi_f(\mathcal{M}\vec{y}) = \prod_{i=1}^{K_Y} \phi_f^{(i)}(\vec{y_i}) \qquad \vec{y_i} \subset \vec{Y} \tag{4.46}$$

The integrand in equation 4.45 specifies a MRF graph G with variable nodes labeled y_1 through y_N. Each of the K_Y subsets $\vec{y_i}$ must be fully connected in G. Additionally, because of the incoming messages $m^t_{j\to f}$, for each row $\mathcal{M}_j$ of $\mathcal{M}$, the variables corresponding to the nonzero entries of $\mathcal{M}_{j*}$ must also be fully connected in G. The computational cost of computing the integral in equation 4.45 with messages represented as histograms will then be $\mathcal{O}(NM^{T_G})$, where T_G is the treewidth of the graph G.

4.3.1 Products of Linear Constraint Nodes

To illustrate the flexibility of this approach, we will now use this analysis to show that messages from a single factor node consisting of a product of K linear experts can be computed in time $\mathcal{O}(NM^{K+1})$. Suppose the potential function ϕ_f over clique $\vec{X}$ is:

$$\phi_f(\vec{x}) = \prod_{k=1}^{K} f_k(\vec{x} \cdot \vec{v}^{(k)}) \tag{4.47}$$

As described in section 4.1.2, one way to implement such a product of multiple linear constraints is by constructing a separate factor node for each constraint f_k (figure 5.1 is an example). Messages from those factors would then be computed independently, each in $\mathcal{O}(NM^2)$ time, using the methods of section 4.1.1. Alternatively, these factor nodes can be combined into one, and the methods of section 4.1.1 no longer apply. The underlying probability distributions represented by these two factor graphs are equivalent; only their factorizations are different. Because belief propagation exploits the structure of the factor graph to perform inference efficiently, the results of belief propagation will depend on the shape of the factor graph even if the underlying probability distribution is unchanged. As mentioned in section 3.3, when sum-product belief propagation converges, the resulting marginals form a minima of the Bethe free energy, a quantity from statistical physics which estimates the distance between the true multivariate probability distribution and the estimated single-variate marginals [103]. The quality of this approximation improves as larger cliques are grouped together [104]. As an extreme example, consider that any probability distribution can be represented by a factor graph with a single factor node connected to each variable node. Inference using belief propagation in such a graph would be exact, but intractable. Conversely, splitting factor nodes into multiple factors typically improves belief propagation efficiency but reduces the quality of the approximation. Thus, combining a product of multiple linear constraints (as in equation 4.47) into a single factor node may cause belief propagation to estimate marginals more accurately than using a separate factor node for each linear constraints. Products of multiple linear constraints within a single factor node are not eligible for the methods of section 4.1.1, but using the transformed variable elimination methods of this section, we can show how these messages can be computed in $\mathcal{O}(NM^{K+1})$ time. Assuming that $N << M$, this represents a computational advantage over the original brute-force approach as long as $K+1 < N$.

Under transformation $\mathcal{T}$, ϕ_f of equation 4.47 becomes

$$\phi_f(\vec{y}) = \prod_{k=1}^{K} f_k(\mathcal{M}\vec{y} \cdot \vec{v}^{(k)}) = \prod_{k=1}^{K} f_k(\vec{y} \cdot \mathcal{M}'\vec{v}^{(k)}) \tag{4.48}$$

where $\mathcal{M}'$ denotes the transpose of $\mathcal{M}$. There are many transforms $\mathcal{T}$ that can reduce the computation of messages from this factor node from $\mathcal{O}(M^N)$ to $\mathcal{O}(NM^{K+1})$. Here, we will choose $\mathcal{M}$ to be an upper-triangular band matrix with bandwidth $K+1$, with row $\mathcal{M}_{1*} = (1, 0, ..., 0)$. Next, we constrain $\mathcal{M}$ so that the vector $\mathcal{M}'\vec{v}^{(k)}$ is zero everywhere except for elements 1 through $K+1$. Note that this ensures that under transform $\mathcal{T}$, in the MRF G underlying Y, y_i and y_j are only connected for $|i-j| \leq K$. This ensures that G has a treewidth of K.

The constraint that the vector $\mathcal{M}'\vec{v}^{(k)}$ is only nonzero in elements 1 through $K+1$ is equivalent to

$$\mathcal{M}_{*i} \cdot \vec{v}^{(k)} = 0 \quad \forall k \leq K, \;\; K+1 < i \leq N \tag{4.49}$$

where $\mathcal{M}_{*i}$ is the i^{th} column of $\mathcal{M}$. By construction, column $\mathcal{M}$ is only nonzero between elements $i-K$ and i. Thus, we can achieve our constraint by setting the $(K+1)$-element vector $(\mathcal{M}_{(i-K),i}, ..., \mathcal{M}_{i,i})$ to be perpendicular to $(v_{i-K}^{(k)}, ..., v_i^{(k)})$ for all $k \leq K$, and $K < i \leq N$. Note that if the bandwidth of $\mathcal{M}$ (and thus the treewidth of G) were any smaller, this constraint could not be satisfied. Also note that for $K = 1$, the transform described here matches the example transform used as an example in section 4.1.1. For the change of variables used in equation 4.9, $\mathcal{M}$ is given by

$$\mathcal{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{v_2} & -\frac{1}{v_2} & 0 \\ 0 & 0 & \frac{1}{v_3} & -\frac{1}{v_3} \\ 0 & 0 & 0 & \frac{1}{v_4} \end{pmatrix} \quad (4.50)$$

$$\mathcal{M}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & v_2 & v_3 & v_4 \\ 0 & 0 & v_3 & v_4 \\ 0 & 0 & 0 & v_4 \end{pmatrix} \quad (4.51)$$

4.3.2 Embedding Additional Potentials

In section 4.1, we mentioned that a good choice of the transform of variables may allow one to embed additional pairwise nonlinear potential functions at no additional cost. We will explain that in more detail here. Suppose that our factorized distribution $P(\vec{X})$ contains the factors $\phi_1(\vec{x})$ and $\phi_2(\vec{x})$, both ranging over the same subset of variables $\vec{x}$, where

$$\phi_1(\vec{x}) = g_1(\vec{x} \cdot \vec{v}) \quad (4.52)$$

$$\phi_2(\vec{x}) = g_2(\vec{x} \cdot \vec{v}_1, \vec{x} \cdot \vec{v}_2) \quad (4.53)$$

One approach is to implement ϕ_1 and ϕ_2 as two separate factor nodes in the factor graph. However, this requires additional computation. Additionally, unnecessarily separating overlapping factors can degrade the Bethe approximation that underlies belief propagation, reducing accuracy [104]. Combining these factors into a single factor node with potential $\phi_1\phi_2$ could be advantageous.

Let $\mathcal{M}$ be a matrix that allows messages $m^t_{\phi_1 \to 1}(x_i)$ from ϕ_1 to variable node x_1 to be computed in $\mathcal{O}(NM^2)$ time (if $\vec{x}$ contains four variables, then $\mathcal{M}$ is the matrix given by equation 4.50). Now suppose that v_1 and v_2 both lie in the plane defined by two consecutive rows j and $j+1$ of $\mathcal{M}^{-1}$. Then, in the transformed space $\vec{y} = \mathcal{M}^{-1}\vec{x}$, the MRF corresponding to ϕ_2 consists of only a single connection joining variables y_j and j_{j+1}. This means that, under the transformed space, the two potential functions ϕ_1 and ϕ_2 have overlapping factor graphs. That allows us to combine ϕ_1 and ϕ_2 into one factor node and still compute messages efficiently.

For example, consider the four-dimensional linear constraint node

$$\phi_1(\vec{x}) = g(v_1x_1 + v_2x_2 + v_3x_3 + v_4x_4) \tag{4.54}$$

discussed in section 4.1. Using the change of variables given by $\mathcal{M}$ in equation 4.50, we can compute messages $M_1 = m_{f\to 1}$ efficiently according to

$$\begin{aligned} M_1(x_1) \propto \int g(v_1x_1 + y_2)\bigg(\int m_2(\frac{y_2 - y_3}{v_2}) \\ \bigg(\int m_3(\frac{y_3 - y_4}{v_3})m_4(\frac{y_4}{v_4})dy_4\bigg)dy_3\bigg)dy_2 \end{aligned} \tag{4.55}$$

Now suppose that $\phi_2(\vec{x}) = h(x_3, x_4)$. Both x_3 and x_4 lie on the plane spanned by $y_3 = v_3x_3 + v_4x_4$ and $y_4 = v_4x_4$. That means that we can represent $\phi_2(\vec{x})$ as

$$\phi_2(\vec{x}) = h(\frac{y_3 - y_4}{v_3}, \frac{y_4}{v_4}) = \hat{h}(y_3, y_4) \tag{4.56}$$

Thus, messages from the combined factor node $\phi_1\phi_2$ can be computed as

$$\begin{aligned} M_1(x_1) \propto \int g(v_1x_1 + y_2)\bigg(\int m_2(\frac{y_2 - y_3}{v_2}) \\ \bigg(\int m_3(\frac{y_3 - y_4}{v_3})m_4(\frac{y_4}{v_4})\hat{h}(y_3, y_4)dy_4\bigg)dy_3\bigg)dy_2 \end{aligned} \tag{4.57}$$

In Chapter 6, this technique will be used for an application that infers 3D shape from a shaded image (see equation 6.10). The approach described here makes it possible to to combine a hard linear constraint that enforced the integrability of the surface:

$$\phi_1(\vec{x}) = \delta((q_1 - q_2) + (p_1 - p_2)) \tag{4.58}$$

with a spatial prior on the second order derivative of depth $\frac{\partial^2 z}{\partial x \partial y}$:

$$\phi_2(\vec{x}) = \exp(-\frac{|q_1 - q_2|}{2b}) \tag{4.59}$$

4.3.3 Sums of Linear Constraint Nodes

It is also useful to note that some potential functions $\phi_f(\vec{x})$ which cannot be made more efficient under any transform $\mathcal{T}$ can be expressed as the *sum* of some number of efficient potential functions. For example, we may find that

$$\phi_f(\vec{x}) = \phi_{f1}(\vec{x}) + \phi_{f2}(\vec{x}) \tag{4.60}$$

where ϕ_{f1} and ϕ_{f2} admit transforms that reduce each potential to a low-treewidth MRF. In such cases, the belief propagation messages $m_{f\to i}(x_i)$ can be computed by summing messages from ϕ_{f1} and ϕ_{f2}:

$$m^t_{f\to i}(x_i) = m^t_{f1\to i}(x_i) + m^t_{f2\to i}(x_i) \tag{4.61}$$

Thus, if a potential is a sum of linear constraint nodes, messages $m_{f\to i}(x_i)$ can be computed in time $\mathcal{O}(bNM^2)$, where b is the number of terms in equation 4.60.

As an example, consider the hard constraint node that enforces that variable x_n is the maximum of several variable nodes:

$$x_n = \max_i\{x_1, \ldots, x_{n-1}\} \tag{4.62}$$

$$\phi_f(\vec{x}) = \delta(x_n - \max_i\{x_1, \ldots, x_{n-1}\}) \tag{4.63}$$

This type of constraint may be useful to extract a pertinent global feature from a stream of variable nodes. The potential $\phi_f(\vec{x})$ can be expressed as a sum of n MRFs with treewidths of 1. To illustrate with $N = 4$:

$$\begin{aligned}\phi_f(\vec{x}) &= \delta(x_4 - \max_i\{x_1, x_2, x_3\}) && (4.64)\\ &= H(x_1 - x_2)H(x_1 - x_3)\delta(x_4 - x_1)\ + \\ &\quad H(x_2 - x_1)H(x_2 - x_3)\delta(x_4 - x_2)\ + \\ &\quad H(x_3 - x_1)H(x_3 - x_2)\delta(x_4 - x_3) && (4.65)\end{aligned}$$

where H is defined by

$$H(x) \equiv \begin{cases} 1 & x > 0 \\ 0 & otherwise \end{cases} \tag{4.66}$$

Each line of equation 4.65 is already in the form of a tree-shaped MRF; no change of variables is needed. Specifically, if we set $\phi_{f1}(\vec{x})$ to be the first line of equation 4.65, then we can compute $m_{f\to i}(x_4)$ as:

$$m^t_{f\to i}(x_4) = m^t_{f1\to i}(x_4) + m^t_{f2\to i}(x_4) + m^t_{f3\to i}(x_4) \tag{4.67}$$

$$m^t_{f1\to i}(x_4) = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{x_1} m_2(x_2) dx_2 \right) \left(\int_{-\infty}^{x_1} m_3(x_3) dx_3 \right)$$
$$\delta(x_4 - x_1) m_1(x_1) dx_1 \tag{4.68}$$

4.4 Message Representation for Belief Propagation

For continuous random variables, the integrals of equation 4.8 or 4.10 typically cannot be computed analytically. In these cases, the beliefs $b_i(x_i)$ and messages $m_{i\to f}(x_i)$ are often approximated by discrete histograms. Discretization error can be a serious issue for histogram representations, especially for highly kurtotic or near-certain beliefs. These errors can propagate across nodes and accumulate over iterations. Also, for some applications, adequate covering of the variable space requires many bins. Even using the techniques of section 4.1, a high value of message length M incurs a high computational cost. In this section, we will discuss different methods of representing messages, and how to reduce discretization error.

4.4.1 Parametric Message Representation

One method of representing belief propagation messages is to assume that each message and belief can be well approximated with a Gaussian [84, 65]. However, for many applications, marginals are often highly non-Gaussian. This is often the case in computer vision, where natural image statistics have distributions which are highly kurtotic. Even more problematic are random variables encoding the hidden underlying parameters of a scene, such as 3D shape or surface material, which often have bimodal or multimodal messages and beliefs. The shape-from-shading application of Chapter 6 and the facial appearance model of [81] are two examples of applications of belief propagation with highly multimodal messages and marginals. Among those problems where the Gaussian approximation is effective, many can be solved more simply using linear programming or gradient descent methods. For these reasons, we will focus here on the more flexible histogram and particle-based representations.

4.4.2 Particle-Based Message Representations

Particle based belief propagation works by approximating each message by a set of samples, or *particles*. Each particle is associated with a mean μ and a weight w. Each message $m_{i\to f}(x_i)$ is represented with M particles, with means $\{\mu_{if}^{(m)}\}_{m=1}^{M}$ and weights $\{w_{if}^{(m)}\}_{m=1}^{M}$. In the case where the potential function ϕ_f is sufficiently simple, such as a small mixture of Gaussians, $m_{f\to i}(x_i)$ can be approximated as:

$$m_{f\to i}(x_i) \approx \tilde{m}_{f\to i}(x_i) = \sum_{m=1}^{M} w_{fi}^{(m)} \phi_f(x_i, \vec{\mu}_{\mathcal{N}(f)\setminus i,f}^{(m)}) \tag{4.69}$$

$$w_{fi}^{(m)} = \prod_{j\in\mathcal{N}(f)\setminus i} w_{jf}^{(m)} \tag{4.70}$$

where $\vec{\mu}_{\mathcal{N}(f)\setminus i,f}^{(m)}$ is a vector composed of the m^{th} particles from each message $\tilde{m}_{j\to f}$ such that $j \in \mathcal{N}(f) \setminus i$ [39]. If ϕ_f is not of a simple form, then it is helpful to perform an additional step where we define $\tilde{m}_{f\to i}(x_i)$ by sampling from equation 4.69, to simplify subsequent computations [81]. In this case, let $\mu_{fi}^{(m)}$ be a sample drawn from $\phi_f(x_i, \vec{\mu}_{\mathcal{N}(f)\setminus i,f}^{(m)})$. We can then approximate $\tilde{m}_{f\to i}(x_i)$ as:

$$\tilde{m}_{f\to i}(x_i) = \sum_{m=1}^{M} w_{fi}^{(m)} \mathcal{N}(x_i; \mu_{fi}^{(m)}, \Lambda_{fi}) \tag{4.71}$$

where $\mathcal{N}(x; \mu, \Lambda)$ is a Gaussian density function with mean μ and variance Λ.

For particle based belief propagation, the computational bottleneck lies in computing $\{\mu_{if}^{(m)}\}_{m=1}^{M}$ according to equation 4.4, which requires sampling from $m'_{i\to f}(x_i)$, defined as:

$$m'_{i\to f}(x_i) = \zeta_{fi}(x_i) \prod_{g\in\mathcal{N}(i)\setminus f} \tilde{m}_{g\to i}(x_i) \tag{4.72}$$

$$\zeta_{fi}(x_i) = \int_{\vec{x}_{\mathcal{N}(f)\setminus i}} \phi_f(\vec{x}) d\vec{x} \tag{4.73}$$

If $\tilde{m}_{g\to i}(x_i)$ is computed as in equation 4.71, this requires sampling from a product of $D - 1$ mixtures of M Gaussians each, where $D = |\mathcal{N}(i)|$.

A straight-forward sampling method would require interpreting $m'_{i\to f}(x_i)$ a weighted mixture of M^{D-1} Gaussians, and sampling from that, which requires $\mathcal{O}(M^D)$ operations. Instead, [81] showed how Gibbs sampling could be used to sample $\{\mu_{if}^{(m)}\}_{m=1}^{M}$ from $m'_{i\to f}(x_i)$ in $\mathcal{O}(D\kappa M^2)$ steps, where κ is the number of Gibbs sampling iterations required. Note that if $\tilde{m}_{f\to i}(x_i)$ is computed as in equation 4.69, this step is made more difficult.

Particle-based belief propagation was originally developed for pairwise connected MRFs using standard belief propagation. Both higher-order potential functions and Heskes' convergent belief propagation pose several additional obstacles for nonparametric belief propagation. Graphs with higher-order cliques tend to be more highly connected, and thus have higher values of D. For instance, the denoising problem in Chapter 5 uses a network with $D = 12$ (see figure 5.1). Particle-based belief propagation is typically considered impractical for $D > 8$ [39].

This problem is exacerbated by the adjustments made in convergent variations of belief propagation. As mentioned earlier, heavily connected graphs typical of problems with high-order cliques, as well as graphs with high-energy potential functions such as hard linear constraint nodes, all tend to benefit greatly from convergent belief propagation. The greatest obstacle to particle-based message representations imposed by convergent belief propagation is the exponentiation of messages, as in equation 3.17. Thus, rather than sampling $\mu_{if}^{(m)}$ from a product of D mixtures of M Gaussians (already a challenging task), samples must be drawn from such a product $r$aised to an arbitrary, fractional exponent. One more difficulty in using particle-based messages for convergent belief propagation is that the potential function ϕ_f has been replaced with $\tilde{\phi}_f$ (in equation 3.18), which requires sampling from a product of more Gaussian mixtures. Thus, for convergent belief propagation, equation 4.72 becomes

$$m'_{i\to f}(x_i) = \zeta_{fi}(x_i)\left(\frac{b_i^{\tau}(x_i)}{\tilde{m}_{i\to f}^{t}(x_i)}\right)^{\frac{n_j-1}{n_j}} \prod_{g\in\mathcal{N}(i)\backslash f} \tilde{m}_{g\to i}^{t-1}(x_i)^{\frac{1}{n_i}} \tag{4.74}$$

Recall that each message $\tilde{m}_{g\to i}^{t-1}$ is represented as a mixture of Gaussians. Sampling from such a distribution would be quite challenging.

4.4.3 Histogram-Based Message Representations

Particle-based representations benefit from the their flexible and dynamic structure, which allow them to focus computational effort on the most likely values of a probability distribution. One way to achieve similar flexibility of representation without incurring the computational expense of sampling from complex distributions is to use histograms with variable-width bins, where each bin may have a different, possibly dynamic, width. Using variable-width bin histograms, messages are approximated as:

$$m_{f\rightarrow i}(x_i) \approx \hat{m}_{f\rightarrow i}(x_i) = \sum_{m=1}^{M} w_{fi}^{(m)} \sqcap_{\beta_i^{(m-1)}}^{\beta_i^{(m)}}(x_i) \tag{4.75}$$

$$\sqcap_{\beta_0}^{\beta_1}(x) \equiv \begin{cases} 1 & x \in [\beta_0, \beta_1) \\ 0 & otherwise \end{cases} \tag{4.76}$$

Variable-width bin histograms have been used successfully to improve the speed and performance of the join tree algorithm [47, 43]. Here we show that such a representation, when applied to belief propagation, can overcome the obstacles encountered in applying particle-based representations to Heskes' guaranteed-convergent LBP variation [29], or to problems with highly-connected graphs. We require that each message $m_{i\rightarrow f}(x_i)$ and $m_{f\rightarrow i}(x_i)$ to and from a given variable node i must have the same bin edges $\{\beta_i^{(m)}\}_{m=1}^{M}$. Because of this, and because histogram bins are non-overlapping (unlike Gaussian kernels), both multiplication and exponentiation now become trivial:

$$\begin{aligned} &\left(\left(\sum_{m=1}^{M} w_m \sqcap_{b_{m-1}}^{b_m}(x_i)\right)\left(\sum_{m=1}^{M} w'_m \sqcap_{b_{m-1}}^{b_m}(x)\right)\right)^{\eta} \\ &\quad = \sum_{k=1}^{M} (w_m w'_m)^{\eta} \sqcap_{b_{m-1}}^{b_m}(x) \end{aligned} \tag{4.77}$$

Thus, equation 3.17 can be computed efficiently, even for high values of D:

$$m_{i\rightarrow f}^{t}(x_i) \approx \hat{m}_{i\rightarrow f}^{t}(x_i) \tag{4.78}$$

$$\hat{m}_{i\rightarrow f}^{t}(x_i) \equiv \hat{m}_{i\rightarrow f}^{t}(x_i)^{\frac{1-n_i}{n_i}} \prod_{g\in\mathcal{N}(i)\backslash f} \hat{m}_{g\rightarrow i}^{t-1}(x_i)^{\frac{1}{n_i}} \tag{4.79}$$

$$= \sum_{m=1}^{M} \left(w_{fi}^{(m)}\right)^{\frac{1-n_i}{n_i}} \prod_{g\in\mathcal{N}(i)\setminus f} \left(w_{gi}^{(m)}\right)^{\frac{1}{n_i}} \sqcap_{\beta_i^{(m-1)}}^{\beta_i^{(m)}}(x_i) \tag{4.80}$$

Using linear constraint nodes, computing messages from factor to variable nodes $m_{f\to i}(x_i)$ (as in equation 4.10) can be viewed as a series of convolutions of scaled histograms. For the example in section 4.1.1, the first step is to compute the integral

$$M_{3,4}(y_3) = \int m_3(\frac{y_3 - y_4}{v_3}) m_4(\frac{y_4}{v_4}) dy_4 \tag{4.81}$$

$$= [m_3(t/v_3) * m_4(t/v_4)](y_3) \tag{4.82}$$

where $*$ denotes convolution. $m_{f\to i}(x_i)$ can be computed as

$$M_{2,3,4}(y_2) = [m_2(t/v_2) * M_{3,4}(t)](y_2) \tag{4.83}$$

$$m_{f\to i}(x_i) = [g(-t) * M_{2,3,4}(t)](-v_1 x_1) \tag{4.84}$$

Consider the simplest case for computing $M_{3,4}(y_3)$, where m_3 and m_4 are represented by histograms with all bins of width 1 ($\beta_3^{(m)} = \beta_4^{(m)} = m$), and $v_3 = v_4 = 1$, so that no scaling is required. Often, such a convolution of histograms is approximated as a discrete convolution:

$$\hat{M}_{3,4}(y_3) = \sum_{m=1}^{M} w_{2,3}^{(m)} \sqcap_{\beta_{2,3}^{(m-1)}}^{\beta_{2,3}^{(m)}}(x_i) \tag{4.85}$$

$$w_{2,3}^{(m)} = \sum_{m'=1}^{M} w_3^{(m-m')} w_4^{(m')} \tag{4.86}$$

$$\beta_{2,3}^{(m)} = m \tag{4.87}$$

However, this approximation can result in compounded discretization error. For example, suppose that $m_3(x) = m_4(x) = \sqcap_0^1(x)$. Then $M_{3,4}(y_3)$ is a piecewise linear function that is nonzero within the interval $(-1, 2)$. However, using the approximation in equation 4.86, $\hat{M}_{3,4}(y_3)$ will be nonzero only within $[0, 1]$, because both $\hat{m}_3$ and $\hat{m}_4$ have only one nonzero bin. A reduction in discretization error can be achieved by discretizing $\hat{M}_{3,4}(y_3)$ *after* the integration is performed:

$$w_{2,3}^{(m)} = \frac{1}{W_m} \int_{\beta_{2,3}^{(m-1)}}^{\beta_{2,3}^{(m)}} \left(\int \hat{m}_3(\frac{y_3 - y_4}{v_3}) \hat{m}_4(\frac{y_4}{v_4}) dy_4 \right) dy_3 \tag{4.88}$$

$$W_m = \beta_{2,3}^{(m)} - \beta_{2,3}^{(m-1)} \tag{4.89}$$

In the more general case, where $\{\beta_i^{(m)}\}_{m=0}^M$ and $\vec{v}$ are all arbitrary, an approximation like equation 4.86 is more difficult. Thus, in general, equation 4.88 is both more accurate and more convenient.

Note that the brute-force $\mathcal{O}(M^N)$ computation of an N dimensional integral of discrete histograms such as equation 4.5 would typically employ a method similar to equation 4.86, where integration is performed *after* discretization. Thus, by using linear constraint nodes, we can reduce discretization error in addition to saving time.

To implement equation 4.88, first observe that the product $\hat{m}_3(\frac{y_3-y_4}{v_3})\hat{m}_4(\frac{y_4}{v_4})$ is equal to a 2D histogram under an affine transform. Equation 4.88 integrates this 2D function over a rectangular region. This is equivalent to summing the areas of a set of four- to six-sided polygons, each weighted by $w_3^{(m-m')}w_4^{(m')}$. It can be shown that the total number of such polygons cannot exceed $3M^2$. Thus, equation 4.88 can be computed in $\mathcal{O}(M^2)$ time.

At the start of the belief propagation algorithm, the locations of histogram bin edges $\{b_m\}_{m=1}^M$ can be initialized based on local estimates of the marginal, such as single-variate potential functions $\phi(x_i)$. In the denoising example in Chapter 5, the intensity value of each pixel has a single-variate Gaussian potential function whose mean is the observed (noisy) pixel intensity. In this case, we set $\{b_m\}_{m=1}^M$ so that each bin is equally likely under this Gaussian distribution.

In some applications, such as the denoising application, it is sufficient to hold these bin widths fixed throughout the belief propagation execution. In other applications, if the range of values x_i is especially large, or if messages are expected to be very low in entropy, then it may be beneficial to adapt the histogram bin edges to best represent the current beliefs. For Heskes' convergent variation of belief propagation, this can be most conveniently done when $b_i(x_i)$ reaches convergence, and $\tilde{\phi}_f$ is updated.

Several strategies are available for adjusting the bin locations of each variable node. One approach is to simply delete low-likelihood bins and split high-likelihood bins apart. This strategy is related to some previous works that adaptively restrict the search space of belief propagation to only those states with high predicted likelihoods [10]. Another strategy is to run a special, single iteration of belief propagation where each bin is first split into 2 or 3 bins. Following this high-resolution iteration, bins can be recombined until only M bins remain. Recombination can be performed to minimize either sum-squared error with the high-resolution message, or the

KL-divergence (as used by [47] to combine two possibly multidimensional histograms). Finally, if messages are expected to approximate a particular functional form, one strategy is to fit the beliefs to some parametric function and place the histogram bins to minimize error. In the denoising application of Chapter 5, a small performance boost can be achieved by placing bins to minimize KL-divergence with a Gaussian fitted to the latest beliefs. Despite the Gaussian arrangement of bin edges, highly non-Gaussian distributions may still be effectively represented by such a histogram. At the same time, placing bins in this way allows belief propagation to focus computational effort on the most likely and most interesting intensity values.

Regardless of the strategy used, if a variable node's bin locations are altered, it is never necessary to perform interpolation to find new values of bin weights $\{w_{fi}^{(m)}\}_{m=1}^{M}$. Beliefs and messages can be retained using the original, unaltered bin locations until the belief propagation algorithm updates that variable node according to equations 4.4 through 4.6. During that update, incoming messages can be constructed using the new bin locations.

Note that the locations of histogram bins for the intermediate messages $\hat{M}_{3,4}$ and $\hat{M}_{2,3,4}$ can also be dynamically adapted. Similar strategies that are available for adapting message bin locations are also available for setting the bin locations of intermediate messages.

4.5 A Particle/Histogram Hybrid Approach

In this section, I propose a hybrid approach to message representation that uses both a histogram and a particle representation to retain the benefits of both systems. As in section 4.4.3, beliefs and messages are first represented as variable-width histograms. All beliefs and messages associated with a particular variable node share the same bin edges β:

$$m_{f\to i}(x_i) \approx \hat{m}_{f\to i}(x_i) = \sum_{m=1}^{M} w_{fi}^{(m)} \sqcap_{\beta_i^{(m-1)}}^{\beta_i^{(m)}}(x_i) \tag{4.90}$$

where the function $\sqcap_a^b(x)$ is 1 on the interval $[a, b]$, and zero elsewhere. This representation allows incoming messages at a variable node to be multiplied together efficiently (as in equation 4.4). It also allows messages to be raised to an arbitrary exponent, thus allowing convergent forms of belief propagation to run efficiently.

The shortcoming of the histogram approach is that for general potential functions, the multi-dimensional summand of equation 4.5 requires $\mathcal{O}(M^N)$ operations. However, particle representations allow this integrand to be computed efficiently using Monte Carlo integration. To achieve the benefits of both representations, we also represent messages using samples. To compute the integral in equation 4.2, we first draw a set of S samples $\{\mu_{jf}^{(s)}\}_{s=1}^{S}$ from each message $\hat{m}_{j\to f}^{t}(x_j)$. Since $\hat{m}_{j\to f}^{t}$ is represented as a histogram, this is both simple and efficient. Now, message $m_{f\to i}(x_i)$ can be approximated by:

$$\tilde{m}_{f\to i}(x_i) = \sum_{s=1}^{S} \phi_f(x_i, \vec{\mu}_{\mathbb{N}(f)\backslash i,f}^{(s)}) \tag{4.91}$$

To avoid expensive Gibbs sampling when exponentiating messages or multiplying messages together, this message is next transformed into histogram form $\hat{m}_{f\to i}(x_i)$:

$$\begin{aligned} w_{fi}^{(n)} &= \int_{\beta_i^{(n-1)}}^{\beta_i^{(n)}} \tilde{m}_{f\to i}(x_i) dx_i && (4.92) \\ &= \sum_{s=1}^{S} \int_{\beta_i^{(n-1)}}^{\beta_i^{(n)}} \phi_f(x_i, \vec{\mu}_{\mathbb{N}(f)\backslash i,f}^{(s)}) && (4.93) \end{aligned}$$

If ϕ_f is simple, it may be possible to compute this integral analytically. Alternatively, methods can be devised to sample from $\phi_f(x_i, \vec{\mu}_{\mathbb{N}(f)\backslash i,f}^{(m)})$, and use the number of samples falling between $\beta_i^{(n-1)}$ and $\beta_i^{(n)}$ to compute $w_{fi}^{(n)}$. For the results in this paper, we approximated the integrals in equation 4.93 by evaluating $\phi_f(x_i, \vec{\mu}_{\mathbb{N}(f)\backslash i,f}^{(m)})$ only at the bin edges and midpoints and integrating numerically using the trapezoid rule. This approximation should be sufficient so long as ϕ_f is smooth, and the histogram bins are narrow.

Using this approach, belief propagation message updates are $\mathcal{O}(DM)$ for messages $m_{i\to f}^{t}(x_i)$ (Eq. 4.4) and $\mathcal{O}(NMS)$ for messages $m_{f\to i}^{t}(x_i)$ (Eq. 4.2). To attain a given level of quality of the estimate $\tilde{m}_{f\to i}^{t}$, the number of samples S must be proportional to the variance of $\phi_f(\vec{x})$.

Note that retaining both histogram and particle representations is not necessary: messages and beliefs can either be stored in histogram form and sampled only for the purpose of integration (Eq. 4.2), or messages can be stored as particles and converted to histograms only for multiplication and exponentiation (Eq. 4.4).

The advantage of the histogram/particle hybrid representation is that, unlike Linear Constraint Nodes, it can be applied to any potential function. The disadvantage is that it is an approximate method, and the computed messages contain more error. By alternating between two approximate message representations, this method is vulnerable to both sampling and discretization error. Therefore, where the use of a single linear constraint node is applicable, that approach is preferable.

Some potential functions can be approximated by the product of several linear constraint nodes, all ranging over a single graph clique. This approach is used by the popular model of natural image priors known as Fields of Experts [73]. Representing such a product of soft linear constraints as a single factor node (rather than seperating each factor into its own factor node) improves the Bethe free energy approximation employed by belief propagation. However, a product of linear constraint nodes is not itself a linear constraint node, and so the computational shortcuts of section 4.1 only apply when each factor is represented using a seperate factor node. The particle/histogram hybrid message representation, however, applies is both cases. Although the hybrid message approach is approximate, the improvement to the Bethe approximation caused by combining factor nodes may overcome the additional noise inherent in the sampling-based integration. For the problem of image denoising using Fields of Experts, this question will be persued more fully in Chapter 5.

4.6 Convergent Belief Propagation

Recall that the original formulation of belief propagation does not always converge. In section 3.4, we mentioned a number of variants of belief propagation that ensure the convergence of the algorithm. In the case of sum-product belief propagation, these methods are guaranteed to converge to a minima of the Bethe free energy. Convergence of regular belief propagation decreases in likelihood when the underlying factor graph has many tight loops or potential functions with high energy (i.e. potential functions that are deterministic or nearly deterministic) [28]. Some of the applications we consider later in the thesis strongly exhibit both of these qualities, and in fact do not converge for regular belief propagation. To overcome this problem we will be using Heskes' double-loop belief propagation algorithm [29]. The use of such variants of belief propagation not only ensure convergence, but they also often result in superior minima of the Bethe free energy in those cases when regular belief propagation does converge [29]. For these reasons, we use this variant of belief propagation throughout the remaining chapters.

As described in section 3.4, there are several variants of belief propagation that ensure convergence, both for sum-product belief propagation **??** and max-product belief propagation **??**. For each of these variants, the computation of belief propagation messages (as in equation 3.6 or 3.11) is a central step of the algorithm. Because the methods of this chapter describe how to efficiently compute belief propagation messages, they are compatible with each of these convergent variants, including Yuille's CCCP [106], the double-loop algorithms of Heskes' et. al. [29], Teh and Wellings UPS [87], the tree-reweighted sum-product belief propagation of Wainwright et. al. [92], and Kolmogorov's tree-reweighted max-product belief propagation [44].

Additionally, the choice of message representation, discussed in section 4.4, can interact significantly with the choice of belief propagation variant. Each of the convergent variants of sum-product belief propagation requires raising messages to some fractional exponent. Taking the exponent of messages that are represented in histogram form is computationally simple and takes only $\mathcal{O}(M)$ time, i.e. linear in the number of histogram bins. Taking the exponent of a message represented by a set of particles is not at all straightforward, and typically requires an expensive resampling method. This is one benefit of the particle/histogram hybrid message representation method discussed in section 4.5.

4.7 Conclusions

In this chapter, we have introduced a way to efficiently perform belief propagation over large graph cliques, reducing computation from $\mathcal{O}(M^N)$ to $\mathcal{O}(NM^2)$ for a wide variety of potential functions. We have shown how these methods can be generalized in several ways to benefit a larger subclass of potential functions. Additionally, we have developed methods for representing belief propagation messages for continuous variables that remain computationally efficient for highly connected graphs, convergent variants of belief propagation, and the use of the computational shortcuts introduced in this paper. These message representations allow discretization error to be minimized while at the same time preserving computational efficiency.

The techniques introduced in this chapter open up a wealth of powerful, higher-order statistical models for inference using belief propagation methods that would previously have been intractable. Belief propagation is a promising framework of optimization for these models, because it often outperforms gradient-based optimization methods by exploiting factorizations, and by performing optimization within the much larger search space of single-variate marginals, which is often less prone to local extrema. Computer vision in particular stands to benefit greatly from higher order statistical models due to the complex statistical structure of natural images and underlying image properties. In chapters 5-7 I will present three applications in computer vision which would have been intractible using standard belief propagation, but can now be made efficient using the methods of this chapter.

Chapter 5

Spatial Priors

Several state-of-the-art computer vision algorithms use belief propagation. A number of these, including stereo [82], photometric stereo [84], shape-from-shading (Chapter 6), image-based rendering [99], segmentation and matting [93] work over a grid at the pixel level. These algorithms solve ambiguous and underconstrained problems, where having a strong prior for images or 3D shape is essential. However, the computational complexity of belief propagation has constrained these algorithms to weak pairwise interactions between neighboring pixels. These pairwise interactions capture the smoothness properties of images, but they overlook much of the rich statistics of natural scenes. Finding a way to exploit a stronger model of image priors using belief propagation could greatly enhance the performance of these algorithms.

One promising recent model for capturing natural image statistics beyond pairwise interactions is the Fields of Experts model (FoE), which provides a way to learn an image model from natural scenes [73]. FoE has shown itself to be highly effective at capturing complex image statistics by performing well at image denoising and image inpainting (filling in holes) using a gradient descent algorithm. The FoE model describes the prior probability of an image as the product of several Student-t distributions:

$$p(\vec{I}) \propto \tilde{p}(\vec{I}) = \prod_{C} \prod_{i=1}^{K} \left(1 + \frac{1}{2}(\vec{I}_C \cdot \vec{J}_i)^2\right)^{-\alpha_i} \tag{5.1}$$

where C is the set of all (overlapping) $n \times n$ patches in the image, and $\vec{J}_i$ is an $n \times n$ filter. The parameters $\vec{J}_i$ and α_i are learned from a database of

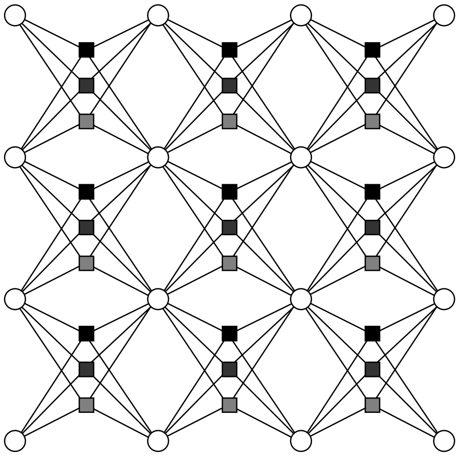


Figure 5.1: A factor graph used to perform image denoising using three 2×2 Fields of Experts filters. Each variable node, shown here as circles, represents the true image intensity at a given pixel. The three gray squares represent factor nodes corresponding to the three 2×2 Fields of Experts filters.

natural images.

Recently, an attempt was made at performing inference in Fields of Experts models using loopy belief propagation, and the approach was tested on an image denoising problem [51]. The authors showed that using three 2×2 Fields of Experts filters yields a significant improvement over pairwise models. In their approach, the authors mitigate the computational complexity of equation 3.6 by restricting the intensity at each pixel to lie within a range defined by its immediate neighbors within the noisy image. Specifically, the true intensity value of each pixel is assumed to lie between the brightest and darkest of its nearest four neighbors within the noisy image, after a slight Gaussian blur is applied. Thus, computational complexity of each message is still $\mathcal{O}(M^N)$, but M (the number of possible labels) is significantly reduced (note that here, $N = 4$). One drawback of this approach is that it is particular to image denoising. In many problems requiring a strong image or range image prior such as stereo and other depth inference algorithms, it can be difficult to restrict the search space of each variable based solely on local properties of the algorithm input. We seek to develop an implementation of Fields of Experts for belief propagation that can be applied to arbitrary image or range image inference problems.

5.1 Fields of Experts using Linear Constraint Nodes

Using the methods of section 4.1, efficient belief propagation is possible in higher-order Fields of Experts factor graphs without relying on simplifying assumptions specific to image denoising. In this section, in order to demonstrate the viability of this approach, we apply our methods to the image denoising problem, using the same 2 × 2 filters as [51]. Although we use image denoising as an example problem, note that this approach is not specific to image denoising, and can be used as a spatial prior for a variety of computer vision applications.

In the denoising problem described here, we are given a natural image (such as figure 5.2**a**) that has been corrupted with additive Gaussian noise of known variance (such as figure 5.2**b**). The object is to remove this noise and recover the original image. Using the Fields of Experts model, the conditional probability of the denoised image $\vec{I}$ given the noisy image $\vec{I}_N$, is modeled by

$$p(\vec{I}|\vec{I}_N) \propto \tilde{p}(\vec{I}|\vec{I}_N) \tag{5.2}$$

$$\tilde{p}(\vec{I}|\vec{I}_N) = \tilde{p}(\vec{I}) \prod_{x,y} \left(\frac{1}{\sigma\sqrt{2\pi}} e^{(\vec{I}(x,y)-\vec{I}_N(x,y))^2/(2\sigma^2)} \right) \tag{5.3}$$

where the (unnormalized) prior $\tilde{p}(\vec{I})$ is the Fields of Experts model given in equation 5.1. The Fields of Experts spatial prior is implemented according to the factor graph in figure 5.1. The Gaussian likelihood is implemented as a prior at each node, and requires no additional messages. Note that this model is capable of performing denoising in a variety of other noise circumstances, such as non-Gaussian or multiplicative noise.

Note that in the factor graph in figure 5.1, the observed, noisy pixel values are not explicitly represented as variable nodes. Instead, the Gaussian likelihood potential functions are absorbed into the factor nodes neighboring each pixel, and therefore require no additional belief propagation messages.

In our implementation, each variable node's beliefs and messages are represented using 16 bins. Bin edges are initialized so that each bin has equal probability under the Gaussian distribution $P(true\_intensity|noisy\_intensity)$, and bins span the range of possible intensity values from 0 to 255. Results are reported for the case where bin edges remain static during the inference procedure, and also for the case where bin edges are occasionally updated

Figure 5.2: (Following page). Using higher-order Fields of Experts to perform image denoising. **a)** A cropping from the original image (from [66]). **b)** The original image with additive Gaussian noise of $\sigma = 20$. **c)** The output of belief propagation over a pairwise-connected Markov Random Field, similar to the model described in [21]. Pairwise models tend to produce piecewise constant image regions [51]. **d)** Denoising using the gradient descent algorithm employed by [73], with three 2×2 Fields of Experts filters learned from natural images. **e)** Results using the same 2×2 FoE model as **d)**, except using linear constraint nodes (the methods described in section 4.1) and the graphical model of figure 5.1. Intensity values were chosen to be the expected value of the estimated marginal. **f)** Results using the same 2×2 FoE model as **d)** and **e)**, except using the partical/histogram hybrid message representation discussed in section 4.5, and the factor graph of figure 5.3.

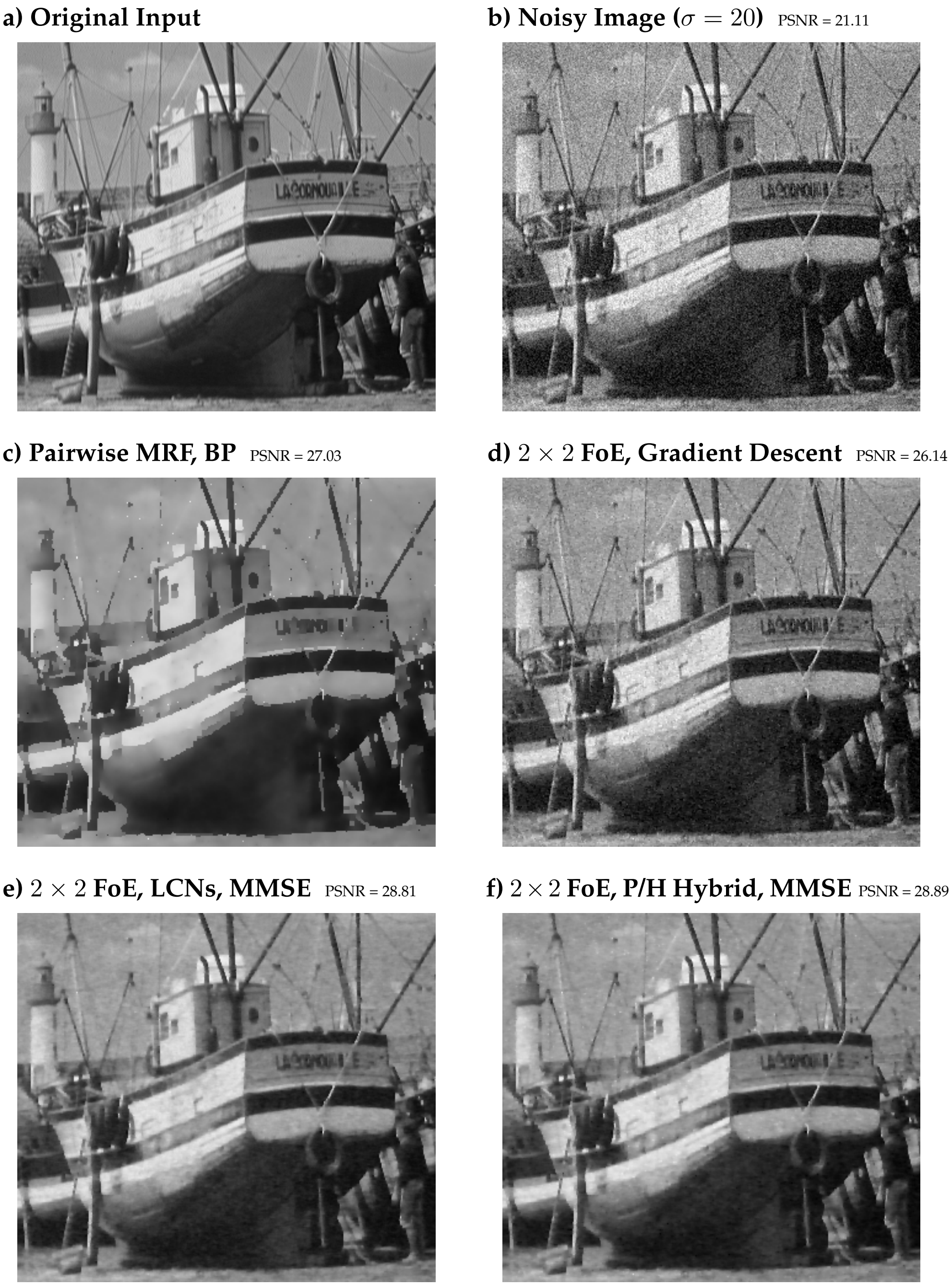
to minimize the KL-divergence between the histogram $\hat{b}_i(x_i)$ and a Gaussian distribution fitted to the current beliefs. Intermediate messages (such as $\hat{M}_{3,4}(y_3)$ and $\hat{M}_{2,3,4}(y_2)$) were represented as histograms with 32 bins. Bin edges for intermediate messages were chosen by first computing a histogram of 96 bins, where edges were chosen to minimize the KL-divergence between the convolution of two Gaussians fit to the two convolved input messages. Then the most unlikely consecutive pairs of these bins were combined until 32 bins remained. We ran each image for 15 outer-loop iterations (15 updates of $\tilde{\phi}_f$, as in equation 3.20) of the convergent belief propagation algorithm described in section 3.4. On average, this required about 35 iterations of belief propagation. The 2×2 Fields of Experts parameters used by our model were the same as those in [51].

For comparison, we also tested 2×2 Fields of Experts using the gradient descent algorithm used in [73] (code available online). In each case, gradient descent was run for 3000 iterations using a step-size of 0.1.

Sample results from our approach are shown in figure 5.2. We measured the average peak signal to noise ratio (PSNR) for each algorithm over the same set of 10 images from the Berkeley segmentation database [61] that was used in [51]. Here, PSNR is defined by

$$PSNR = 20 \log_{10}(255/\sqrt{MSE}) \tag{5.4}$$

where MSE is the mean squared error. These results are shown in table 5.1. In tables 5.1 and 5.1, we also show results for five canonical images from denoising literature, as used by Portilla [66].



a) Original Input

b) Noisy Image ($\sigma = 20$) PSNR = 21.11

c) Pairwise MRF, BP PSNR = 27.03

d) 2×2 FoE, Gradient Descent PSNR = 26.14

e) 2×2 FoE, LCNs, MMSE PSNR = 28.81

f) 2×2 FoE, P/H Hybrid, MMSE PSNR = 28.89

(Caption on Previous Page)

	MAP		MMSE	
	$\sigma = 10$	$\sigma = 20$	$\sigma = 10$	$\sigma = 20$
Noisy Input Images	28.13	22.11	28.13	22.11
Hand-tuned Pairwise MRF using belief propagation [51]	30.73	26.66	NA	NA
2×2 FoE using gradient descent (algorithm from [73])	30.59	26.09	NA	NA
2×2 FoE using belief propagation (from [51])	30.89	27.29	NA	NA
2×2 FoE using LCNs, Fixed Histograms	31.41	27.12	31.51	27.29
2×2 FoE using LCNs & Adaptive Histograms	31.55	27.25	31.62	27.40
2×2 FoE using Particle/Histogram Hybrids	31.72	27.52	31.79	27.66

Table 5.1: Peak signal-to-noise ratio (PSNR), in decibels, for pairwise and higher-order models, averaged over the ten images from the Berkeley segmentation database [61] used in [51]. PSNR is defined in equation 5.4. For each belief propagation algorithm, a MAP point estimate is approximated by choosing the maximal value of each marginal, and a MMSE point estimate is taken by computing the mean of each marginal. Denoising using linear constraint nodes (LCNs) with 2×2 FoEs outperforms both belief propagation on pairwise MRFs and gradient descent on identical FoEs. Using particle/histogram hybrid message representations (section 4.5) results in a small additional performance gain.

As shown in figure 5.2**c**, belief propagation over pairwise-connected Markov random fields tends to produce piecewise constant results. A 2×2 Fields of Experts model promises to correct this problem by modeling not only the statistics of neighboring pixels, but whole blocks of four pixels. However, gradient descent (fig. 5.2**d**) is unable to fully exploit this model, achieving signal-to-noise ratios that do not exceed those of the pairwise-connected model using belief propagation. Local minima encountered by gradient descent are likely at fault. Figures 5.2**e** and 5.2**f** show the results of our approach, which outperform both pairwise connected MRFs and gradient descent over the same statistical model (2×2 FoEs) by over a decibel of PSNR.

More importantly, using the methods of section 4.1, belief propagation can be performed efficiently in higher-order factor nodes without relying on domain-specific approximations or simplifying assumptions. On a 2.2GHz Opteron 275, our algorithm takes under two minutes for each iteration of belief propagation on a 256×256 image. By comparison, the method of [51] took 16 minutes per iteration on a 3GHz Xeon, and benefited from a reduced search space.

In addition to an improvement in running time, our approach also yielded some improvement in quality over the more brute-force belief propagation approach used by Lan et. al. [51]. One difference between these two methods is that, in order to reduce the search space of the problem, [51] relies on

$\sigma = 10$ (PSNR = 28.13)	boat	peppers	house	lena	barbara
2x2 FoE, Gradient Descent [73]	30.61	30.73	31.00	30.91	30.19
2x2 FoE, BP using LCNs, Fixed Histograms	32.30	32.84	33.70	33.35	30.43
2x2 FoE, BP using LCNs, Adaptive Histograms	32.28	32.85	33.71	33.34	30.24
2x2 FoE, Particle/Histogram Hybrid	32.32	33.03	33.88	33.46	30.54
5x5 FoE, Gradient Descent [73]	33.04	34.18	35.14	35.03	32.85
Portilla et. al. [66]	33.58	33.77	35.35	35.61	34.03

Table 5.2: Denoising results for five canonical denoising images (used in [66]). Image noise $\sigma = 10$. *BP using LCNs* refers to belief propagation using the linear constraint node computational shortcut. State of the art denoising algorithms (bottom two rows) are also reported. Note that these algorithms are designed especially for denoising, and would be difficult to use as a spatial prior for other vision tasks like stereo, shape from shading, and others. All error values are given in peak signal-to-noise ratio (equation 5.4). For each belief propagation algorithm, only the MMSE point estimates are given. Maximum marginal estimates were typically similar or slightly worse.

$\sigma = 20$ (PSNR = 22.11)	boat	peppers	house	lena	barbara
2x2 FoE, Gradient Descent [73]	26.14	26.15	26.49	26.45	25.44
2x2 FoE, BP using LCNs, Fixed Histograms	28.77	28.98	30.53	30.30	25.93
2x2 FoE, BP using LCNs, Adaptive Histograms	28.81	29.09	30.46	30.31	25.47
2x2 FoE, Particle/Histogram Hybrid	28.89	29.29	30.53	30.25	25.77
5x5 FoE, Gradient Descent [73]	29.82	30.19	32.02	31.81	28.31
Portilla et. al. [66]	30.38	30.31	32.39	32.66	30.32

Table 5.3: Results as in table 5.3, except under noise with $\sigma = 20$. All error values are given in peak signal-to-noise ratio (equation 5.4).

the assumption that pixel intensities in the original image should lie within some range determined by their immediate neighbors in the noisy image. Because this assumption is only applicable for image denoising, and our interest lies in developing spatial priors that can be used by any belief propagation algorithm, our approach does not restrict the search space. This assumption is violated by just under 10% of the pixels in the images tested, so it is reasonable to ask if this assumption could account for the improvement in performance achieved by our approach. However, when our linear constraints node algorithm is forced to make this same assumption and restrict the search space for each pixel according to its neighbors, performance *improves* slightly. For the suite of 10 Berkeley images tested in table 5.1, restricting the search space as in [51] increased PSNR from 31.62 to 31.68 for $\sigma = 10$ and from 27.40 to 27.57 for $\sigma = 20$. This improvement most likely results from focusing histogram bins on more likely intensity values. However, while this assumption may offer some small performance gain, it is important to remember that such assumptions are not available for other applications of spatial priors where belief propagation is more necessary, such as stereo [82], photometric stereo [84], shape-from-shading (Chapter 6), image-based rendering [99], segmentation, and matting [93].

If the assumption made by Lan et. al. [51] to reduce the search space is not the cause of the performance gain of our approach, then it is most likely due to the convergent variant of belief propagation [28] and nonparametric message representations used by our approach. One reason that this performance gain is of interest is that although the underlying statistical model of natural images is identical between the two methods, the factor graph used by [51] (shown in figure 5.3) is not identical to the one used by our method (shown in figure 5.1). The graph used in [51] uses a single factor node for all three 2×2 experts within a clique, whereas our method separates each expert into its own factor node. By separating out these factors, the Bethe free energy approximation used by belief propagation is degraded. The good performance of our approach shows that this sacrifice in the quality of the Bethe approximation was less than the advantages offered by convergent belief propagation and variable width bin histograms. In section 5.2, I will show how the particle/histogram hybrid message representation introduced in section 4.5 can be used to perform efficient belief propagation in this original, unseparated factor graph (figure 5.3). This results in a small improvement in PSNR.

For the sake of comparison, we also present results for two state-of-the-art denoising algorithms: 5×5 FoEs using gradient descent [73], and an algorithm that uses Gaussian scale mixtures to model the joint distribution of wavelet coefficients [66]. These algorithms are designed specifically for image denoising; they cannot easily be adapted for use as spatial priors in more complex algorithms like stereo, shape from shading, matting, and others. We present them here in tables 5.2 and 5.3, for a sense of perspective.

Belief propagation computes the single-variate marginals of each pixel value. The expected value of the denoised image, or the minimum mean-squared error (MMSE) point estimate, can be computed by taking the mean of each marginal. This approach usually yields the best results for our algorithm. In table 5.1 we also show results for the intensity values that maximize the marginal, or the "maximum marginal" (MM) point estimate. For fixed-width histograms, a continuous MRF that approximates intensity using only 16 bins would typically show high discretization error for point estimates computed this way. By using variable width histograms, these quality of these point estimates is nearly equal to MMSE results. As discussed in section 3.4, maximum a posteriori (MAP) point estimates can be computed using either non-convergent max-product belief propagation, or by performing annealing within convergent sum-product belief propagation [107]. For problems with smooth, unimodal likelihood functions like image denoising, using MAP point estimates is rarely beneficial.

In table 5.1, results using linear constraint nodes are presented both with and without dynamic readjustment of histogram bin locations. In each case, histogram bins are initialized so that each bin has an equal likelihood under to the Gaussian likelihood function. In the dynamic case, bins are also adjusted after each outer-loop iteration, as described earlier. This procedure takes negligible time, and yields a small but significant performance improvement. For other applications, where initial estimates of the marginals may be less accurate, or beliefs fluctuate more during inference (such as the shape-from-shading algorithm described in Chapter 6), dynamic histogram bin edge adjustments are more important to performance.

In addition to showing that LCNs allow belief propagation to efficiently capture nonpairwise aspects of the statistics of natural scenes, we are also interested in showing that belief propagation outperforms gradient descent techniques at finding maximally likely images I that optimize $\tilde{p}(\vec{I}|\vec{I}_N)$ (equation 5.3). In tables 5.4 and 5.5, we show the unnormalized log likelihoods $\log \tilde{p}(\vec{I}|\vec{I}_N)$ for the denoised images computed by both gradient descent and by our belief propagation approach. These algorithms both use the same 2×2 Fields of Experts model, and so both algorithms are attempting to opimize the same energy function. Because the spatial prior $P(\vec{I})$ may not be optimal, it is possible for an algorithm to achieve poor denoising results dispite finding superior optima of $\tilde{p}(\vec{I}|\vec{I}_N)$. Tables 5.4 and 5.5 show that this is not the case. All variants of belief propagtion with LCNs find denoised images that are significantly more likely (according to the model) than those chosen by gradient descent.

$\sigma = 10$	Berkeley Suite	boat	peppers	house	lena	barbara
Original Image	-3.93	-26.62	-6.57	-6.40	-25.82	-27.84
Noisy Image	-4.15	-28.25	-7.03	-6.92	-27.78	-29.25
2x2 FoE, Gradient Descent [73]	-3.94	-26.65	-6.65	-6.52	-26.20	-27.79
2x2 FoE, BP using LCNs, Fixed Histograms	-3.80	-25.70	-6.45	-6.31	-25.35	-26.73
2x2 FoE, BP using LCNs, Adaptive Histograms	-3.80	-25.68	-6.44	-6.31	-25.33	-26.69
2x2 FoE, Particle/Histogram Hybrid	-3.79	-25.64	-6.43	-6.30	-25.29	-26.68

Table 5.4: The (unnormalized) log-likelihood of each image reconstruction according to the 2×2 FoE model. All values are given as $\log \tilde{p}(\vec{I}|\vec{I}_N) \times 10^{-5}$, where $\tilde{p}(\vec{I}|\vec{I}_N)$ is given in equation 5.3. The values given for the Berkeley Suite images show the mean unnormalized log-likelihood for the ten images from the Berkeley segmentation database [61] used in [51]. The four denoising algorithms shown here all seek to optimize the same equation (i.e. equation 5.3 using the 2×2 FoE model). In each case, belief propagation significantly outperforms gradient descent. Thus, in addition to producing denoised images with less error, belief propagation does a better job at finding the optimum values of the FoE probability model. This means that the improvement in performance is not due to peculiarities of the FoE model. Also note that, according to the model, the denoised images computed using belief propagation have greater likelihood than the original image. This suggests that improving the model is now more important than improving the method of optimization.

5.2 Fields of Experts using Particle/Histogram Hybrid Representations

In section 5.1, I showed how, using the linear constraint nodes shortcut of section 4.1, belief propagation could be made both efficient and effective for graphical models that exploit the Fields of Experts spatial prior. In order for the linear constraint node shortcut to be applicable, the three 2×2 experts used here needed to be represented as three separate factor nodes, as in figure 5.1. Each of these factor nodes is a soft linear constraint node, and thus available for the computational shortcuts of section 4.1. As mentioned earlier, separating these factors into separate factor nodes does not affect the probability distribution that is represented by the factor graph. However, belief propagation is an approximate method, and the Bethe approximation intrinsic to belief propagation is more accurate when the factors are combined into one factor node per image patch, as in figure 5.3.

The histogram/particle hybrid representation introduced in section 4.5

$\sigma = 20$	Berkeley Suite	boat	peppers	house	lena	barbara
Original Image	-4.19	-28.44	-7.03	-6.85	-27.64	-29.66
Noisy Image	-4.86	-33.39	-8.33	-8.25	-33.10	-34.04
2x2 FoE, Gradient Descent [73]	-4.29	-29.29	-7.33	-7.21	-28.93	-30.18
2x2 FoE, BP using LCNs, Fixed Histograms	-4.00	-27.89	-7.00	-6.87	-27.55	-28.88
2x2 FoE, BP using LCNs, Adaptive Histograms	-3.99	-27.85	-6.99	-6.86	-27.52	-28.82
2x2 FoE, Particle/Histogram Hybrid	-3.99	-27.79	-6.98	-6.84	-27.46	-28.82

Table 5.5: Results as in table 5.4, except under noise with $\sigma = 20$. All values are given as $\log \tilde{p}(\vec{I}|\vec{I}_N) \times 10^{-5}$, where $\tilde{p}(\vec{I}|\vec{I}_N)$ is given in equation 5.3.

can be applied to any potential function. However, unlike the linear constraint node computational shortcut, the histogram/particle hybrid representation is approximate in the sense that it introduces additional error into the belief propagation messages beyond the unavoidable discretization error. Thus, while this technique can be applied directly to the factor graph of figure 5.3, it is unclear whether improvement in the Bethe approximation outweighs the error caused by sampling-based integration.

In this section, the histogram/particle hybrid representation was used to perform belief propagation in the Fields of Experts denoising factor graph of figure 5.3. We used the same FoE model as in section 5.1, which uses three 2×2 linear experts. As in section 5.1, messages in histogram form contained 16-bin, and bin edges were occasionally updated to minimize the KL-divergence between the histogram $\hat{b}_i(x_i)$ and a Gaussian distribution fitted to the current beliefs. Messages in particle form were represented using 50 samples per message.

Results are shown in figure 5.2 and listed in tables 5.1 through 5.5. The results using hybrid message representations show a small improvement over the separated linear constraint node approach, suggesting that, in this case, the approximation error of Monte Carlo integration was less than the improvement made to the Bethe approximation. Increasing the number of particles did not improve the results significantly, suggesting that the difference in performance reflects the sacrifice in quality of the Bethe approximation made by separating linear constraints into multiple factor nodes. In this example, that difference appears to be small but not negligible.

It is important to point out that this result can be expected to vary between different applications. The amount of error introduced by using Monte Carlo integration techniques depends both on the number of samples used and also on the variance of the integrand:

$$\phi_f(\vec{x}) \prod m^t_{j\to f}(x_j)$$

In the present application, the potential functions f is a smooth product of three Student-t distributions, and the messages m tend to have a simple, near-Gaussian, unimodal form. In this application, increasing the number of particles did not significantly alter the results, suggesting that sampling error was small. In other applications with more complex potentials, sampling error could have been a much greater issue. The difference in the quality of the Bethe approximation between the factor graphs of figures 5.1 and 5.3 is more difficult to anticipate, but it can be expected to depend on number of factors and the similarity between each potential function.

In the denoising application described here, computing each message using a histogram/particle hybrid representation takes slightly longer than using Linear Constraint nodes. However, because the hybrid representation allows us to represent all three linear features in a single factor node, the hybrid representation reduces the number of messages that must be computed. In a 240×160 image, denoising required 57 seconds per iteration on a 3GHz Xeon, versus 96 seconds using Linear Constraint nodes. Convergence typically required approximately 30 iterations.

5.3 Conclusions

A great many problems in computer vision aim to infer properties of real scenes under highly ambiguous and underconstrained circumstances. For these problems, an accurate probabilistic prior is paramount for success. Many of these same problems also require sophisticated methods of statistical inference. Approaches such as gradient descent and related methods often have difficulty finding solutions on these more difficult problems of computer vision. Belief propagation has provided a successful tool for solving many of these problems, such as stereo [82], photometric stereo [84], shape-from-shading (Chapter 6), image-based rendering [99], segmentation and matting [93]. Because of the intense computational demands of belief propagation through higher-order cliques, previous applications of belief propagation were limited to pairwise-connected spatial priors. These pairwise priors often fail to capture the rich statistical structure of real images. In this chapter, I show how linear constraint nodes and particle/histogram hybrid message

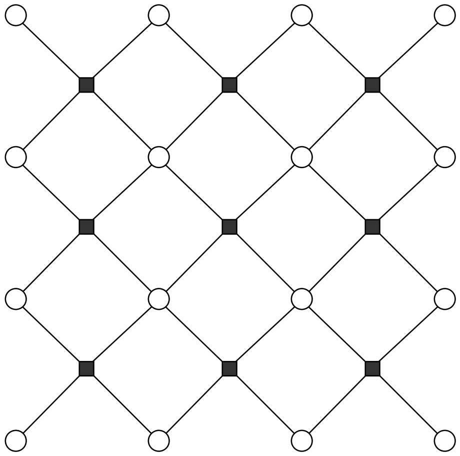


Figure 5.3: A factor graph for performing image denoising using three 2×2 Fields of Experts filters, where (unlike that of figure 5.1), each of the three experts is combined into a single factor node (black square) for each image patch. This factor graph is used in section 5.2 using the particle/histogram hybrid message representation technique introduced in section 4.5. This is also the factor graph used by Lan et. al. [51].

representations can make higher-order spatial priors, like Fields of Experts, efficient. This advance promises to improve the inference results of a number of computer vision applications.

To illustrate this technique, I apply belief propagation to the problem of image denoising. Image denoising makes a good benchmark for the performance of a spatial prior because it uses a simple Gaussian likelihood function and it is easy to reproduce and compare results for different priors and different inference techniques. Because of its simple Gaussian likelihood function, image denoising can be solved efficiently using gradient descent and other approaches. However, these approaches cannot generalize to applications with more complex likelihood functions such as stereo, shape-from-shading, image-based rendering, segmentation, and matting. Our goal is to develop methods of exploiting powerful higher-order spatial priors using belief propagation, which retains the ability to generalize to these more difficult visual inference problems.

The linear constraint node techniques of Chapter 4 make it possible for us to exploit the popular Fields of Experts spatial prior efficiently using belief propagation. When applied to image denoising, linear constraint nodes produced a sizeable speed increase over a previous approach to implement FoE using belief propagation [51], and our approach had the additional benefit that efficient inference did not come at the cost of sacrificing the ability to generalize from image denoising to more complex problems of visual inference (such as stereo or shape-from-shading).

As expected, belief propagation using the higher-order spatial priors of the FoE model produced a significant improvement over belief propagation in the hand-tuned pairwise MRFs of [21]. This improvement offered by higher-order spatial priors may be of great benefit to a number of computer vision tasks that seek to infer images or range images in ambiguous, uncertain circumstances, including stereo [82], photometric stereo [84], shape-from-shading (Chapter 6), image-based rendering [99], segmentation, and matting [93].

It is also important to note that belief propagation using FoE significantly outperformed gradient descent using the same FoE spatial prior. This confirms that, even for a smooth Gaussian likelihood function like that of image denoising, where gradient descent methods can be expected to perform well, belief propagation still offers a significant advantage.

Chapter 6

Shape From Shading

Shape-from-shading (SFS) is a classic computer vision problem that has been studied since photometric investigations of the lunar surface were performed in the 1920s [35]. The goal of SFS is to recover the 3D surface shape given a single image, where all light comes from a single, known direction, and the surface is assumed to have a Lambertian (matte) reflectance and constant albedo (no surface markings or colorations). Under these conditions, the image can be computed from the 3D surface according to the Lambertian equation. Let $N = (p, q, 1)$ be the surface normal vector, and let $S = (p_s, q_s, 1)$ be the known illumination vector. Then the Lambertian equation can be written:

$$i(x, y) = \max(0, \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}}) \qquad (6.1)$$

Here we leave out known quantities of albedo and illumination strength. Note that $p = \frac{\partial z}{\partial x}$, $q = \frac{\partial z}{\partial y}$, where $z(x, y)$ is the surface depth-map. Because our image is spatially discrete, we approximate these as $p(x, y) = z(x+1, y) - z(x, y)$ and $q(x, y) = z(x, y+1) - z(x, y)$.

Our task is to try to invert this computation, to estimate z from i. One reason that this is difficult is that the inverse is a highly nonlinear partial differential equation. Another difficulty is that SFS is highly underconstrained: for any given image i, there are many possible 3D surfaces z which satisfy equation 6.1.

In figure 6.1, we show the factor graph that we propose for solving this problem. This graph uses an overcomplete representation of surface shape: for each pixel, there is a variable node for both p and q. Because the representation is overcomplete, there are linear dependencies among the variables.

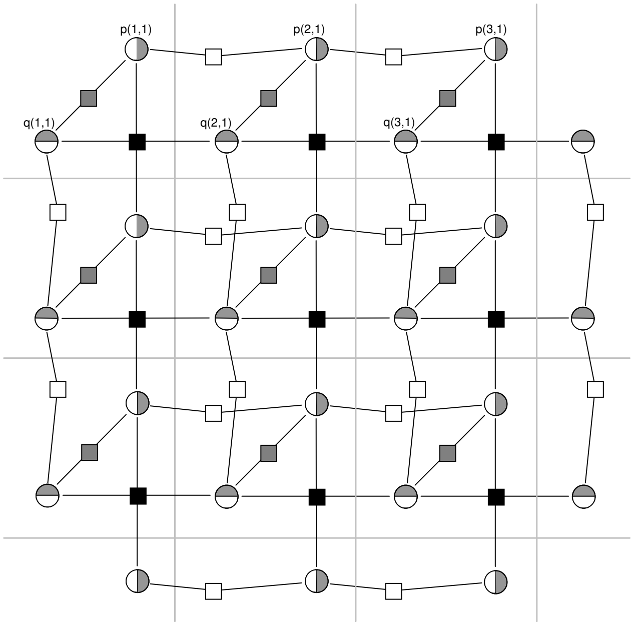


Figure 6.1: Shape-from-shading factor graph for a 3×3 image. Variable nodes are shown as circles, and factor nodes as shown as squares. Variable nodes include nodes for $p = \frac{\partial z}{\partial x}$ and $q = \frac{\partial z}{\partial y}$. Factor nodes include Lambertian constraint nodes (gray), integrability constraint nodes (black), and smoothness nodes (white). Light gray lines indicate the borders between pixels.

Specifically, an identity holds that

$$p(x, y) - q(x, y) + q(x + 1, y) - p(x, y + 1) = 0 \tag{6.2}$$

Failure to enforce these linear dependencies results in internally inconsistent surface normals that violate the zero curl requirement, and thus do not integrate to form a valid 3D surface. Recall that the zero curl requirement states that

$$\frac{\partial}{\partial y}\left(\frac{\partial z}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{\partial z}{\partial y}\right) \tag{6.3}$$

Satisfying these constraints has historically been problematic for SFS. Using the methods of section 4.1, we can enforce these linear dependencies efficiently using hard linear constraint nodes. These integrability constraint nodes are shown in figure 6.1 as black squares. These integration nodes are similar to

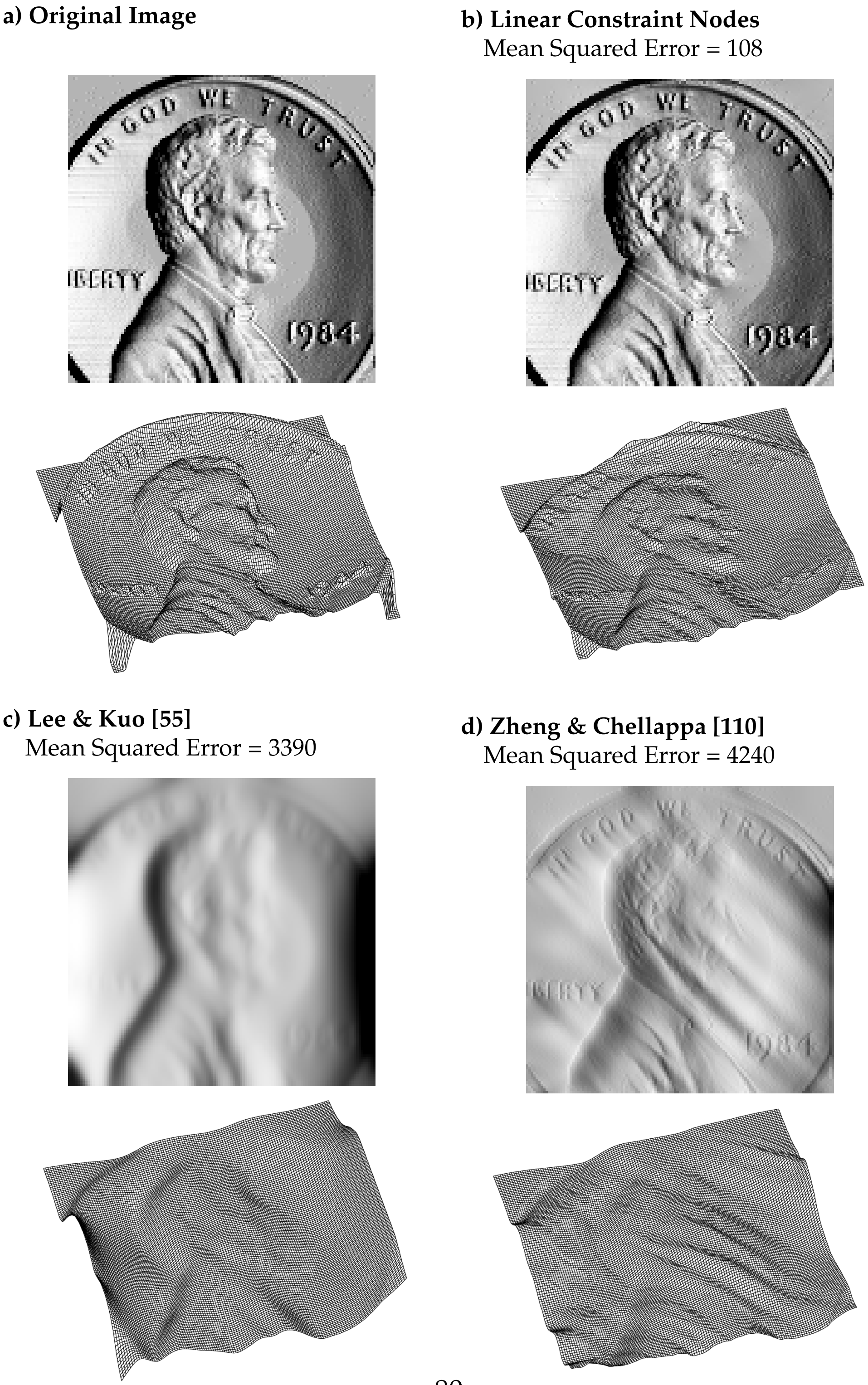
Figure 6.2: (Following page). Comparing our SFS results (column b) with previous energy-minimization approaches (columns c & d). Each subfigure contains a 3D wire mesh plot of the surface (bottom) and a rendering (top) of that surface from a light source at location $(1, 0, 1)$, using the Lambertian reflectance equation. **a)** The original 128×128 surface [109]. The rendering in this column serves as the input to the SFS algorithms in the next three columns. 1001 pixels in this image lie in black shadow. **b)** The surface recovered using our linear constraint node approach. Good results (image MSE < 226) were achieved in under 3 hours, the results in column b were run to convergence (MSE $= 108$ in 24 hours). **c)** The surface recovered using the energy minimization method described by Lee and Kuo [55]. This algorithm performed best out of six SFS algorithms reviewed in the survey paper [109]. **d)** The surface recovered using the method described by Zheng and Chellappa [110] (which performed second-best in [109]). Our approach (column b) offers a significant improvement over previous energy-minimization methods. It is important to note that re-rendering the surface output from our algorithm closely resembles the original input image (the mean squared error of each re-rendering is listed above each image). This means that the Lambertian constraint at each pixel was satisfied, and that any error between the original and recovered surface is purely the fault of the model of the prior probability of natural 3D shapes that was used (in this case, only smoothness was used). The code for the algorithms shown in **c** and **d**, as well as the test image, were acquired through the authors of [109].

those used in [65], except that here, the nonlinear nature of the SFS problem prevents us from approximating the marginals and messages at each variable as Gaussians. In fact, the marginals at each variable are often highly bimodal. Thus, the methods of section 4.1 are required to perform belief propagation at these nodes efficiently.

The square nodes shown in gray in figure 6.1 represent Lambertian constraint nodes. The potential function at these nodes is defined to be the joint likelihood of p and q given image intensity i: $\phi_L(p, q) = P(p, q|i)$. Here, we define $P(p, q|i) = \mathit{const}$ whenever equation 6.1 holds, and zero otherwise.

$$\phi_L(p, q) = \delta\left(i(x, y) - \max(0, \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}})\right) \tag{6.4}$$

An example of a potential function $\phi_L(p, q)$ was shown in figure 2.4. Note that, while we restrict ourselves to the Lambertian equation for this example, any reflectance function could have been used to generate the potentials



a) Original Image

b) Linear Constraint Nodes
Mean Squared Error = 108

c) Lee & Kuo [55]
Mean Squared Error = 3390

d) Zheng & Chellappa [110]
Mean Squared Error = 4240

(Caption on Previous Page)

$\phi_L(p,q)$. By substituting non-Lambertian reflectance functions, these potentials could easily be changed to handle specular surfaces, or even scenes with multiple or diffuse light sources. Furthermore, specifying nondeterministic potentials $\phi_L(p,q)$ would allow us to perform inference when surface reflectance, surface albedo, or lighting conditions are uncertain. This feature is not typical of SFS formulations.

Shape from shading is a highly underconstrained problem. For any input image, there exist many different 3D surfaces that render to the same image under identical lighting. To see this, note that for a $n \times n$ image, there are $2n(n+1)$ variable nodes, but only n^2 Lambertian constraints and n^2 integrability constraints. That leaves $2n$ unconstrained dimensions. Each surface within this large subspace is a valid solution. Additionally, for each pixel that lies in shadow, the Lambertian constraint becomes an inequality at that pixel, and so the number of degrees of freedom increases. In the 128×128 penny image in figure 6.2, 1001 pixels lie in black shadow. Thus, the set of range images that re-render to an image identical to the input image defines a space with up to 1257 dimensions. Any point in this subspace satisfies both the Lambertian and the integrability constraints.

Many shape from shading methods handle this ambiguity by assuming that the surface shape is known along the image border [109]. More recently, Prados [71] has developed methods to recover the computing the maximal surface, where each point is as close to the observer as possible. The assumption that the surface is maximal can be interpreted as a type of spatial prior, although it cannot be learned or adjusted, and it is not clear to what extent real 3D scenes mirror this assumption. A bigger disadvantage of this method is that it is limited to work only when the scene contains no attached shadows; typically this restricts the light source to lie very close to directly behind the camera. Both of these approaches to resolving ambiguity work by introducing additional assumptions on scene parameters and further restricting the types of images in which the algorithm can work.

Even without any additional constraints, SFS is already a highly restricted problem domain, and is applicable only in highly specialized cases. One goal of this thesis is to offer techniques to broaden, rather than constrict, the subclass of scenes that can be approached using shape-from-shading techniques.

A more flexible and robust approach to solving underconstrained problems is to learn or define a probabilistic shape prior $p(z)$ that reflects the likelihood that a given surface shape might occur in nature. Then we can select the 3D shape that maximizes this prior while still rendering to the original input image. In shape-from-shading, this approach is known as energy-minimization (e.g. [55], [110]). Unfortunately, due to the nonlinear nature of the problem, local minima are a serious issue that have prevented energy-minimization approaches from achieving adequate results [109]. Belief propagation methods have proven themselves more robust to local minima, which makes belief propagation a promising new approach to shape from shading.

Here, the 3D surface priors used here are modeled by a product of Laplace distributions:

$$p(Z) \propto \prod_{p,q} \exp(-\frac{|p| + |q|}{\sigma_1}) \exp(-\frac{|\partial p/\partial x| + |\partial p/\partial y| + |\partial q/\partial y|}{\sigma_2}) \tag{6.5}$$

Priors of the form $\exp(-\frac{|p|}{\sigma_1})$ and $\exp(-\frac{|q|}{\sigma_1})$ can be absorbed into the factor nodes adjacent to each variable, and so they require no additional message passing. Priors of the form $\exp(-\frac{|\partial p/\partial x|}{\sigma_2})$ and $\exp(-\frac{|\partial q/\partial y|}{\sigma_2})$ are implemented using a set of pairwise-connected factor nodes. These nodes are shown as white squares in figure 6.1.

Priors of the form $\exp(-\frac{|\partial p/\partial y|}{\sigma_2})$ can be embedded into the integrability hard linear constraint nodes at no additional computational cost. Specifically, define the potential function for this clique as

$$\begin{aligned} \phi_I(p_1, p_2, q_1, q_2) &= \delta\left((p_1 - p_2) - (q_1 - q_2)\right) f(p_1 - p_2) && (6.6) \\ &= \delta\left((p_1 - p_2) - (q_1 - q_2)\right) f(q_1 - q_2) && (6.7) \\ f(x) &\equiv \exp(-\frac{|x|}{\sigma_2}) && (6.8) \end{aligned}$$

then using the same change of variables used in equations 4.11 - 4.14, we can compute outgoing messages as

$$\begin{aligned} M_{p1}(p_1) &= \iiint \phi_I(p_1, p_2, q_1, q_2) m_{p2}(p_2) m_{q1}(q_1) m_{q2}(q_2)\, dp_2\, dq_1\, dq_2 && (6.9) \\ &\propto \int \delta(p_1 + y_2) \Bigg(\int m_{p2}(y_3 - y_2) f(y_3) \\ &\quad \Bigg(\int m_{q1}(y_4 - y_3) m_{q2}(y_4) dy_4 \Bigg) dy_3 \Bigg) dy_2 && (6.10) \end{aligned}$$

where the change of variables is now:

$$y_2 = -p_2 + y_3 \quad = -p_2 - q_1 + q_2 \tag{6.11}$$

$$y_3 = -q_1 + y_4 \quad = -q_1 + q_2 \tag{6.12}$$

$$y_4 = +q_2 \qquad = +q_2 \tag{6.13}$$

This technique allows us to include within a linear constraint node of potential $f(\vec{v} \cdot \vec{x})$ one or more additional potential function $f'(\vec{v}' \cdot \vec{x})$ where v'_i is either equal to v_i or zero. This method is very useful at both reducing the number of messages that must be computed per iteration, and also at improving the Bethe approximation implicit in belief propagation.

In figure 6.2, we show the results of our SFS model. We also compare our results with two previous energy-minimization methods [55, 110]. These methods were the top two SFS algorithms studied in the 1999 survey [109]. Our approach offers a significant improvement over these methods. Further, notice that the surface recovered by our method, when re-rendered under the original lighting conditions, resembles the original input image almost exactly. This means that our approach is able to find a 3D surface that satisfies both the Lambertian equations and the integrability constraints. Of those surfaces that satisfy these constraints, the algorithm is able to select one that is considerably more "likely" than the original ground-truth 3D surface, according to the surface prior model in equation 6.5. Further improvement to the results of this approach can only be achieved by improving the model of the 3D surface priors (equation 6.5).

Note that the model of the 3D surface priors used in this section use the same weak pairwise form that we improve upon in Chapter 5. An obvious next step for this model is to learn Field of Experts filters for 3D surfaces, and then apply these to our SFS model using the methods of Chapter 5. In Chapter 7, we apply the efficient belief propagation methods of Chapter 4 to the problem of learning MRF parameters, and then use this training method to learn effective spatial priors for 3D shape that can be exploited by the SFS approach shown here. Such a spatial prior might also be highly useful for stereo [82], photometric stereo [84], and other forms of depth inference.

6.1 Conclusions

When convergent variations of belief propagation were first introduced, it was unknown whether graphical models that diverge under traditional belief propagation could be satisfactorally solved by forcing convergence. One common belief was that, for cases where belief propagation did not converge, the Bethe approximation was likely to be poor anyway. Several studies had suggested that graphical models for which traditional belief propagation does not converge would give poor results for parameter values or initial condition where convergence is reached [97]. As stated by Heskes [27] in his paper on convergent belief propagation methods,

> "Whether double-loop algorithms are worth the effort is an open question ..."

The success of the shape-from-shading application given here should put this question to rest. Even using a variety of different scheduling, dampening, and reweighting procedures, the model shown here does not converge under traditional belief propagation, even for very simple input conditions. Furthermore, we found that graphical models containing integrability nodes typically diverge even when the initial conditions are correct but perturbed only slightly; that is, at least some of the minima found by belief propagation are unstable minima under traditional belief propagation. Thus, convergent methods belief propagation are required for the SFS approach shown here. It is not clear how large is the class of problems for which traditional belief propagation does not converge but for which the Bethe approximation remains viable, but it is now known that this class includes some important real-world applications.

One of our primary goals in developing a statistical approach to solving SFS was that a statistical approach should be generalizable in new ways to more natural scenes and conditions. In addition to improved performance on the classical SFS problem, the belief propagation approach shown here has a very unique potential to generalize to less restrictive depth inference scenarios.

First, as mentioned earlier, the methods described here are not limited to Lambertian reflectance. The potential function ϕ_L used by the Lambertian constraint nodes could easily be replaced by any other reflectance function. In the more general case, we can define

$$\phi_L(p,q) = \delta\left(i(x,y) - R(p,q,\vec{L},\vec{V})\right) \tag{6.14}$$

where $\vec{L}$ is the lighting vector and $\vec{V}$ is the viewing angle (typically defined to be $(0,0,1)$). Thus, the inference of shape from shading can proceed even with more realistic surface material qualities, such as surfaces with some specular component. In many previous approaches to SFS, the exact form of the Lambertian reflectance function is intimately hardcoded into the inference algorithm, so that altering the reflectance function of the surface is difficult or impossible. Developing methods of SFS that are flexible in this way is an important advantage.

Additionally, the classic formulation of shape-from-shading requires that the scene is lit by only a single light source. This restriction can also be relaxed in our approach. Again, by altering our potential function $\phi_L(p,q)$, arbitrary lighting arrangements can be accommodated. Let illumination be defined by a function $\mathcal{L}(\vec{L})$ that gives the radiance of light coming in from direction $\vec{L}$. Then we can generalize $\phi_L(p,q)$ by:

$$\phi_L(p,q) = \delta\left(i(x,y) - R(p,q,\mathcal{L},\vec{V})\right) \tag{6.15}$$

This allows for shape inference under multiple point light sources, diffuse light sources, or other more natural arrangements. This level of flexibility is very unusual among previous SFS approaches.

The statistical approach to SFS given here also allows us to handle uncertainty in the illumination or surface reflectance properties. In the above examples, the potential function ϕ_L has been deterministic, in the sense that ϕ_L is constant when the surface normal is consistent with the shaded image, and zero otherwise. The use of non-deterministic forms of ϕ_L would not affect the computational requirements of belief propagation, and in fact, lower-energy potentials may be expected to improve the convergence rate [28]. In cases where the exact location of the light source is only known approximately, or the exact reflectance properties of the surface is only approximately known, some fuzziness in the potential function ϕ_L may address that uncertainty. In a very simple approach, we may consider defining

$$\phi_L(p,q) = f\left(i(x,y) - R(p,q,\mathcal{L},\vec{V})\right) \tag{6.16}$$

where f is some loss function, such as a Gaussian $f(x) = \exp(-\frac{x^2}{2\sigma^2})$. Alternatively, we could imagine computing a distribution over $R(p,q,\mathcal{L},\vec{V})$ given a distribution over $\mathcal{L}$ or over R. The ability to perform inference under uncertainty can help bring shape-from-shading closer to handling more natural scenes.

As mentioned earlier, another benefit of the belief propagation approach to SFS shown here is its ability to exploit strong models of spatial priors. In the example in figure 6.2, the Lambertian and integrability constraints are almost exactly satisfied. The 3D shape that is computed by the algorithm is smoother (according to the simple, ad hoc Laplace-based spatial prior of equation 6.5) than the original ground-truth 3D shape. The only way to appreciably improve the output in this example is to improve the spatial prior exploited by the algorithm. The flexible framework of belief propagation over arbitrary factor graphs makes this easy to do. Higher order spatial priors, such as the Fields of Experts model discussed in Chapter 5, can be incorporated into this algorithm simply by adding connections to the existing smoothness factors of figure 6.1. In Chapter 7, I address this possibility further.

Finally, the flexible form of the statistical approach to SFS facilitates the combination of multiple depth cues. For example, the ability to combine shading and stereo cues has long been sought after [14, 68], since the limitations of the two cues are thought to complement each other: stereo excels at computing coarse shape features over a limited depth range, while shading works best at finding fine spatial features over an arbitrary depth range. Stereo has been successfully solved using belief propagation for MRFs in the past [82]. In fact, currently, the top seven stereo algorithms, as evaluated according to the Middlebury stereo evaluation, all use belief propagation [78]. Combining the SFS factor graph of figure 6.1 with the underlying graphs of these models should be straight-forward.

Other depth cues can be exploited in this framework as well. Cues that inform the surface normal at a point, such as texture gradients, perspective cues, and object-border cues, can all be implemented as pairwise factors over the p and q node at a pixel, with the potential function $\phi(p, q|image)$ conditioned on the image. Such potentials could be combined with the current Lambertian potentials $\phi_L(p, q)$, which means that these extra cues would require no additional computation. The inference of 3D shape from texture or perspective cues has historically been carried out in a deterministic setting. However, a probabilistic approach can be handled straight forwardly, by encoding texture gradients and off-parallel line cues as a set of local image features Γ, and then defining a pairwise potential $\phi(p, q|\Gamma)$ that depends on these features. Object border cues inform us that, along occlusion contours, the surface normal of the nearer object must lie perpendicular to the viewing angle. One simple implementation of such a cue would simply require increasing the likelihood of oblique angles near strong image edges. A more sophisticated approach to locating occlusion contours would be to explicitly model occlusion edge locations within the factor graph, using techniques like those described by Hoiem et. al. [32].

Chapter 7

Learning

We now have a collection of methods that allow belief propagation to perform efficient inference for continuous-valued MRFs with higher-order cliques. In this chapter, we show how we can leverage these efficient inference methods into efficient parameter learning techniques. Our goal in this chapter is not only to improve the efficiency of parameter learning, but also to develop a method of parameter learning that is especially suited to inference using belief propagation. In section 7.1, we describe further our goals for a learning method specifically suited to inference using belief propagation.

7.1 Motivation

Ambiguity is a very common problem faced by computer vision applications; for any image or video input, there is often no unique interpretation. When inferring 3D shape, scene segmentations, occlusion contours, surface material types, or one of many other image properties, our goal is to choose the *maximally likely* interpretation. The shape-from-shading application of Chapter 6 is one example of this: for any given 2D image, many 3D surfaces will render, under identical lighting conditions, to match that image. To resolve this ambiguity, we must rely on strong spatial priors to weed out unlikely 3D shapes and select the most likely 3D interpretation. In the example in figure 6.2, both the Lambertian and the integrability constraints were met almost exactly; the only way to significantly improve the resulting shape output is to improve the model for the prior probability of 3D shapes, $P(Z)$. A number of visual tasks are similarly underconstrained and ambiguous, such as stereo, image super-resolution, novel scene synthesis, segmentation, and matting.

One other property that these visual problems have in common is complexity. the statistical relationships found in natural images and scenes are rich, complex, and often difficult to exploit mathematically. For such problems, simpler optimization techniques often cannot be applied. As seen in the shape-from-shading example, approaches like gradient descent often struggles with local minima. Many of the applications listed above have been approached using belief propagation with highly successful results, such as stereo [82], photometric stereo [84], super-resolution [23], segmentation and matting [93], and shape-from-shading (Chapter 6).

The ability to exploit rich spatial priors using belief propagation is therefore very important. In Chapter 5, we showed how linear constraint nodes could be used to allow belief propagation applications to exploit three 2×2 Fields of Experts (FoE) priors. We showed that image denoising using these spatial priors produced a significant improvement over hand-tuned pairwise-connected spatial priors. We also showed that belief propagation found a significantly better maximum likely point estimate for the 2×2 FoE model than gradient descent.

Recall that Fields of Experts models the prior probability of an image as the product of Student-t distributions:

$$p(\vec{I}) \propto \prod_{C} \prod_{i=1}^{K} \left(1 + \frac{1}{2}(\vec{I}_C \cdot \vec{J}_i)^2\right)^{-\alpha_i} \tag{7.1}$$

where C is the set of all (overlapping) $n \times n$ patches in the image, and $\vec{J}_i$ is an $n \times n$ filter [73]. In the original FoE paper, Roth and Black learned 24 5×5 FoE filters. Denoising performed using gradient descent with these large filters produces results that were comparable with the current state-of-the-art [73, 66]. As shown in tables 5.2 and 5.3, using gradient descent, the 24 5×5 outperform the 3 2×2 filters substantially, by about three to four decibels.

The ability to exploit these larger filters using belief propagation would be a great benefit to the many underconstrained applications that rely on belief propagation mentioned above. However, each of the 5×5 FoE filters that performed successful denoising in [73] correspond to a MRF with cliques of size $N = 25$ and $D = 25$ factors per variable node. Even using the efficient inference techniques described previously, belief propagation would be quite slow for such a network. In order to develop a statistical prior that works efficiently under belief propagation, we must be more careful in our use of resources.

In this chapter, our goal is to develop a method for learning spatial priors that will complement the specific requirements of belief propagation, by squeezing the greatest benefit possible from small to moderate sized graph cliques. We accomplish this in two ways. First, the original Fields of Experts model assumed that the Student-t distribution is the best form for potential functions over the linear features of images. This is based on the observation that the empirical marginals of image features tend to resemble Student-t distributions. However, there is no guarantee that potential functions should resemble the empirical marginals; in fact the relationship between them is highly complex, and it requires expensive learning algorithms to estimate potential functions from an empirical distribution. Studies by Zhu and Mumford [111] using lengthy Gibbs sampling procedures have shown that learning arbitrary potential functions (represented by discrete histograms) for similar MRF models could result in non-trivial, and even inverted, potentials. Our learning method relaxes the Student-t assumption, allowing arbitrary potential functions to be learned over image features.

Secondly, most learning methods work by finding values for model parameters Θ that minimize the KL-divergence between the model distribution $P(\vec{X}|\Theta)$ and the empirical distribution $P_0(\vec{X})$. However, in the applications we discuss here, finding the exact values of the model distribution is computationally intractable; instead, we must rely on approximate methods such as belief propagation. Belief propagation computes beliefs $b_i(x_i)$ that approximate the marginals of $P(\vec{X}|\Theta)$. Specifically, the beliefs computed by belief propagation minimize the Bethe free energy $D_{bethe}(\{b_i\}||P(\vec{X}|\Theta))$ (equation 3.13). Since it is these beliefs that are the output of belief propagation, it is the beliefs $b_i(x_i)$, and not the exact marginals of $P(\vec{X}|\Theta)$, that we wish to resemble the empirical distribution P_0. Our method seeks to minimize the distance between the *estimated* marginals $b_i(x_i)$ and the empirical distribution P_0. Thus, our learning procedure attempts to compensate for the approximation inherent in the belief propagation procedure.

Finally, because our method takes advantage of the efficient belief propagation methods described in Chapter 4, it is highly efficient, even for higher order, non-pairwise factor nodes.

7.2 Methods of Learning MRF Parameters

Suppose we have a factorized probability distribution of the form:

$$p(\vec{X}) = \frac{1}{Z(\Theta)} \prod \phi_i(\vec{x_i}, \Theta) \qquad \vec{x_i} \subset \vec{X} \tag{7.2}$$

$$Z(\Theta) = \sum_{\vec{X}} \prod \phi_i(\vec{x_i}, \Theta) \tag{7.3}$$

where Θ is some vector that contains the parameters of the model. Here, $Z(\Theta)$ is a normalization factor that causes $p(\vec{X})$ to sum to one, also known as the *partition function.* In the past, the objective for learning parameters Θ has typically been to maximize the log likelihood (according to the model) of the empirical data:

$$L = \log \left(\prod_{s=1}^{S} p(\vec{X}^{(s)}) \right) \tag{7.4}$$

where $\vec{X}^{(s)}$ is one of S empirical datapoints. In our examples, empirical datapoints will be natural images or natural range images.

The most common technique for learning MRFs for natural image priors is Hinton's Contrastive Divergence [31]. Contrastive Divergence is a way of performing gradient descent on the log likelihood L of the model. This gradient descent provides the update rule:

$$\delta\theta_i = \eta \left(\mathbf{E}\left[\frac{\partial}{\partial\theta_i} \log p(\vec{x}) \right]_{p_0} - \mathbf{E}\left[\frac{\partial}{\partial\theta_i} \log p(\vec{x}) \right]_{p} \right) \tag{7.5}$$

where p_0 is the distribution of the empirical data, and p is the model distribution (as in equation 7.2). Traditionally, the rightmost term must be approximated by using Gibbs sampling to generate a dataset of "fantasy" images from the model distribution p. Gibbs sampling can be slow to converge. Contrastive Divergence improves this approach by taking only one or two iterations of Gibbs sampling. However, learning using Contrastive Divergence for many parameters remains very computationally demanding.

Several statistical models of natural images can be trained using Contrastive Divergence. One popular example is the Fields of Experts (FoE) model [73] discussed above and in Chapter 5. Both the original 5×5 experts and the 2×2 experts used in Chapter 5 were trained using Contrastive Divergence. Since the FoE model was developed, several additional techniques have been developed for learning the FoE filters J and Student-t parameters α [96, 85]. The assumption that the optimal potential functions can be approximated using Student-t distributions is typically retained, because learning a potential function with many parameters would be computationally expensive.

One important subclass of MRF models is the log-linear, or *maximum entropy* (ME) model. ME models are MRFs where each potential function in equation 7.2 can be expressed as $\phi_i(\vec{x}_i, \Theta) = \exp(\theta_i f_i(\vec{x}_i))$. For ME models, equation 7.5 simplifies to

$$\delta\theta_i = \eta \left(\mathbf{E}[f_i(\vec{x})]_{p_0} - \mathbf{E}[f_i(\vec{x})]_p \right) \tag{7.6}$$

Thus, when the parameters Θ are properly trained, the model's marginals of each feature f_i must match those of the empirical distribution. Additionally, of all probability distributions that share this property, the trained ME model will achieve the maximum possible entropy [112]. Intuitively, this suggests that the ME model makes as few assumptions as possible regarding features that the model was not trained on.

Parameters for ME models can be trained by performing gradient descent using equation 7.6 directly. Another popular approach is Generalized Iterative Scaling (GIS) [16], which updates parameters according to the rule

$$\delta\theta_i = \frac{1}{L} \left(\log(\mathbf{E}[f_i(\vec{x})]_{p_0}) - \log(\mathbf{E}[f_i(\vec{x})]_p) \right) \tag{7.7}$$

where $L = \sum_i f_i(\vec{x})$. Both GIS and gradient descent require computing the marginals of the features $f_i(\vec{x})$ with respect to the model distribution. For MRFs with higher-order cliques, computing these marginals has historically been very expensive. Typically, computationally demanding Gibbs sampling must be used to generate sample data from the model. However, using the Linear Constraint Nodes techniques of Chapter 4,these marginals can now be computed much more efficiently.

When the Student-t potential functions of FoE are generalized to arbitrary discrete histograms, it becomes the Minimax Entropy of Zhu, Wu, and

Mumford [112]

$$p(\vec{I}) \propto \prod_{C} \prod_{i=1}^{K} f_i(\vec{I}_C \cdot \vec{J}_i) \tag{7.8}$$

$$= \exp\left(\sum_{C} \sum_{i=1}^{K} \sum_{j=1}^{M} \theta_{i,j} \sqcap_{\beta_i^{(m-1)}}^{\beta_i^{(m)}} (\vec{I}_C \cdot \vec{J}_i) \right) \tag{7.9}$$

For fixed filters J, the Minimax Entropy model is a ME model. Originally, the parameters $\theta_{i,j}$ were trained using a demanding Gibbs sampling procedure, and the filters J were selected from a predetermined set of candidates. Later, Coughlan and Yuille applied GIS to learn the parameters of pairwise-connected ME models (using derivative filters J with a support size of two pixels) by using CCCP, a convergent form of belief propagation, to estimate the filter marginals [12].

For learning arbitrary potential functions for FoE or minimax entropy models, the GIS update equation (equation 7.7) can be simplified. For the i^{th} linear filter $\vec{J}_i$, let $\hat{o}_i(x_i)$ be the empirical marginal of that linear filter, and let $b_i(x_i)$ be the CCCP or LBP-approximated marginal of that linear filter. Then the GIS update equation simplifies to:

$$\delta f_i(x_i) = \frac{1}{K} \left(\log(b_i(x_i)) - \log(\hat{o}_i(x_i)) \right) \tag{7.10}$$

where K is the number of potential functions to be learned.

The use of belief propagation to estimate the marginals used during the GIS procedure was given a formal justification by Teh and Welling [87]. These authors sought to solve a problem they call "generalized inference", which is to estimate the marginals of a factorized probability distribution $P(\vec{X}) = \prod \phi_i(\vec{x_i})$ while the marginals at some subset of nodes V is held fixed. Specifically, they sought to compute a distribution $Q(\vec{X})$ (or the marginals of $Q(\vec{X})$) such that Q minimizes

$$Q = \underset{Q'}{\operatorname{argmin}}\, KL(Q'||P) \tag{7.11}$$

subject to the constraint that

$$Q(x_i) = \hat{o}_i(x_i) \qquad \forall i \in V \tag{7.12}$$

for some set of empirical marginals $\hat{o}_i(x_i)$.

When V is empty, generalized inference reduces to the more standard inference of the form discussed in this thesis: computing the marginals of a factorized distribution. Generalized inference can also be seen to be a generalization of the learning problem. It can be shown that Q will have the same factorization as P, so that

$$P(\vec{X}) = \prod_{i=1}^{K} \phi_i(\vec{x_i}) \qquad Q(\vec{X}) = \prod_{i=1}^{K} \phi'_i(\vec{x_i}) \qquad \vec{x_i} \subset \vec{X} \tag{7.13}$$

Thus, solving for Q requires computing the potential functions ϕ'_i that make the marginals $Q(x_i)$ match the empirical distributions $\hat{o}_i(x_i)$. For example, the problem of learning arbitrary potential functions for a FoE or Minimax Entropy spatial prior model can be posed as a generalized inference problem.

One approach to solving generalized inference is to use GIS to learn the potential functions for Q. However, as described above, GIS requires computing the marginals of Q at each iteration, which can be very expensive. When the marginals of Q cannot be computed exactly, belief propagation might be used. However, the impact of using approximate marginals on the GIS algorithm was unknown. One possibly more principled approach to the generalized inference problem is to minimize the Bethe free energy between a set of estimated marginals (or beliefs) $\{b_i(x_i)\}$ and factorized distribution P subject to the constraints that $b(x_i) = \hat{o}_i(x_i) \forall i \in V$. Teh and Welling showed that the beliefs that minimize the Bethe free energy are equivalent to the beliefs that would be computed by GIS if belief propagation were used to estimate the marginals of Q at each iteration. Since the problem of generalized inference is both a generalization of the inference problem and of the learning problem, this finding provides an additional theoretical justification for using belief propagation to estimate the marginals during GIS.

Using the computational shortcuts described in Chapter 4, we are now able to efficiently use GIS and gradient descent methods (Eq 7.6) to learn Minimax Entropy model potential functions in non-pairwise MRFs. In addition to permitting efficient training, this also allows us to learn arbitrary potential functions for linear image features J without relying on a Student-t assumption. Our goal is to maximize the benefit of MRFs with cliques of moderate size, which can then serve as efficient spatial priors using belief propagation in a wide range of computer vision algorithms. In section 7.3, we describe how training MRFs with GIS and belief propagation can be further refined to learn potential functions that are optimized specifically for inference with LBP by compensating for the approximation that is inherent in LBP inference.

7.3 Optimizing MRFs for LBP Inference

Historically, methods for learning the parameters of MRF models have worked by minimizing the KL divergence between the model distribution $P(\vec{X}|\theta)$ and the empirical distribution $P_0(\vec{X})$. When inference is performed using belief propagation, the marginals of $P(\vec{X}|\theta)$ cannot be computed exactly; they can only be approximated. Thus, to minimize the error of belief propagation inference, we should instead minimize the difference between the empirical marginals and the *approximate* marginals computed by belief propagation.

When MRF parameters are learned by applying GIS with LBP-estimated marginals, it is these approximate marginals, and not the exact marginals, that are compared with the empirical marginals. The fixed points of GIS (see equation 7.10) only occur where the approximate marginals match the empirical marginals exactly. When marginals can be computed exactly, GIS is guaranteed to converge to a fixed point of equation 7.10. However, in the applications we will discuss in this chapter, when using LBP-estimated marginals, GIS typically fails to converge, even when using very low learning rates η. One reason that this might happen is that there may be *no* parameters Θ such that the LBP-estimated marginals match the empirical marginals. When that is the case, the GIS algorithm with LBP-estimated marginals will have no fixed points.

One possible solution is to consider guaranteed-convergent solutions to the approximate generalized inference problem of Teh and Welling [87]. Teh and Welling introduced such a convergent algorithm for generalized inference called Unified Propagation and Scaling (UPS) that uses an approach similar to the tree-based reparametrization techniques for convergent belief propagation discussed in section 3.4 [92, 44]. Unfortunately, in the event that no parameters Θ will cause the LBP-estimated marginals $b_i(x_i)$ to match the empirical marginals $\hat{o}_i(x_i)$, UPS will not satisfy our original goals for a learning algorithm. Ideally, we would want our learning procedure to find potential functions so that the LBP-approximated marginals are *as close as possible* to the empirical marginals. However, generalized inference *constrains* the inference procedure to force the estimated marginals $b_i(x_i)$ to match the empirical marginals $\hat{o}_i(x_i)$ for nodes i in V. In the event that no parameters Θ make

$b_i(x_i)$ match $\hat{o}_i(x_i)$, UPS will *alter* the results of the inference process, so that the marginals estimated by UPS will not match the marginals estimated by LBP. Since we will be performing inference with LBP, this is not ideal.

To solve our learning problem directly; to find MRF parameters Θ that caused LBP-approximated beliefs $b_i(x_i)$ to match empirical marginals $\hat{o}_i(x_i)$ as closely as possible, we would first choose some metric $M(\{b_i(x_i)\}||\{\hat{o}_i(x_i)\})$ between two sets of marginals. This could be a sum of KL-divergences, or a sum-squared error metric. Then, we would minimize this metric with respect to Θ. If the approximated marginals were known to be exact, i.e.

$$b_i(x_i) = \sum_{j \neq i} \sum_{x_j} P(\vec{X}|\Theta) = \sum_{j \neq i} \sum_{x_j} \sum_{k} \exp(\theta_k f_k(\vec{X})) \tag{7.14}$$

then b_i would vary continuously with respect to Θ, and so we could differentiate $M(\{b_i(x_i)\}||\{\hat{o}_i(x_i)\})$ with respect to Θ. However, in general, $b_i(x_i)$ will not be exact. In that case, the LBP-approximated beliefs are the minima of the Bethe free energy:

$$\{b_i\} = \operatorname*{argmin}_{\{b'_i\}} D_{bethe}(\{b'_i\}||P(\vec{X}|\Theta) \tag{7.15}$$

Unfortunately, because D_{bethe} is not convex, the minima of D_{bethe} is not guaranteed to vary continuously with Θ. In fact, for the applications discussed in this chapter, such as learning arbitrary potential functions for FoE image priors, we have observed that the LBP-approximated beliefs sometimes change quite dramatically for very small changes in Θ. This discontinuity makes minimizing $M(\{b_i(x_i)\}||\{\hat{o}_i(x_i)\})$ (which depends on the minima of D_{bethe}) much more challenging.

When attempting to optimize a non-differentiable function, the typical approach is to use derivative-free optimization techniques such as Powell's method or the Melder-Need downhill simplex method. However, the dimensionality of our search problem is problematic for these techniques; our search-space has KM dimensions, where K is the number of potential functions to be learned, and M is the number of histogram bins for each potential function. Methods like Powell's method or the Melder-Need method would require KM evaluations just to initialize the search algorithm. Because each evaluation requires waiting for belief propagation to converge, this would be highly expensive.

One way to overcome the high computational expense of the standard derivative-free optimization methods is to exploit the fact that the gradient of the data log-likelihood (equation 7.6) and the GIS update equation (equation 7.10) can be used as approximate derivatives for minimizing $M(\{b_i(x_i)\}||\{\hat{o}_i(x_i)\})$. In the next two sections, our approach to finding parameters $\Theta = \{f_i(x_i)\}$ that minimize M will work using this idea. Each iteration will begin by updating the potential functions Θ according to a dampened GIS update

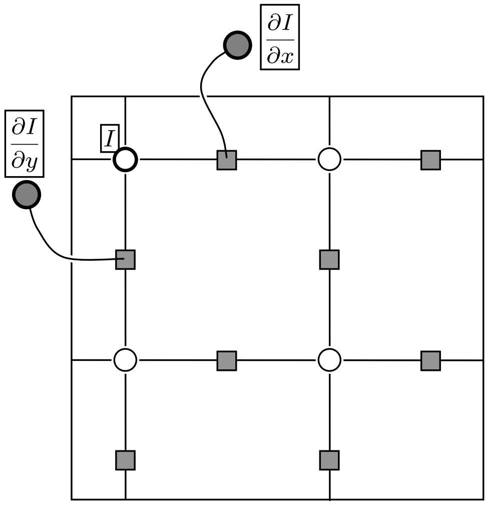
$$f_i^{t+1}(x_i) = f_i^t(x_i) + \frac{1}{\eta K}\left(\log(b_i^t(x_i)) - \log(\hat{o}_i(x_i))\right) \qquad (7.16)$$

where η is some dampening coefficient, and $f_i^t(x_i)$ denotes the i^{th} potential function at iteration t. Then, LBP will be run using the updated potential functions $f_i^{t+1}(x_i)$ to compute the new beliefs $b_i^{t+1}(x_i)$. Next, we evaluate $M(\{b_i^{t+1}(x_i)\}||\{\hat{o}_i(x_i)\})$. If M has increased, meaning that the LBP-approximated marginals are less similar to the empirical marginals than at iteration t, then the dampening coefficient η will be increased and f_i^{t+1} will be recomputed from equation 7.16. This forces the procedure to converge, and ensures that the resulting potential functions $f_i(x_i)$ will result in LBP-approximated beliefs $b_i(x_i)$ that resemble the empirical marginals $\hat{o}_i(x_i)$ as closely as GIS was able to find.

7.4 Results for Image Denoising

We now demonstrate the use of efficient belief propagation to learn MRF parameters by training spatial priors for natural images. To compute the marginals $\mathbf{E}[f_i(\vec{x})]_p$ in equations 7.6 and 7.7, we use belief propagation within a small MRF. To avoid boundary conditions, the borders of the MRF are connected cyclically, as on a torus. As long as the MRF is larger than the image features (to prevent nodes from passing messages to themselves), the size of this torus does not affect the computed marginals. Thus, a 2×2 grid is sufficient in our case. In general, we can use the Linear Constraint node techniques of Chapter 4 for cliques that hold only a single potential function, and the particle/histogram hybrid approach (section 4.5) for cliques that contain multiple experts. The ground-truth marginals $\mathbf{E}[f_i(\vec{x})]_{p_0}$ were computed offline from the 200 training images of the Berkeley segmentation database [61].

Recently, empirical studies have found that second-order gradient descent methods, such as L-BFGS, applied to equation 7.6 often outperform GIS

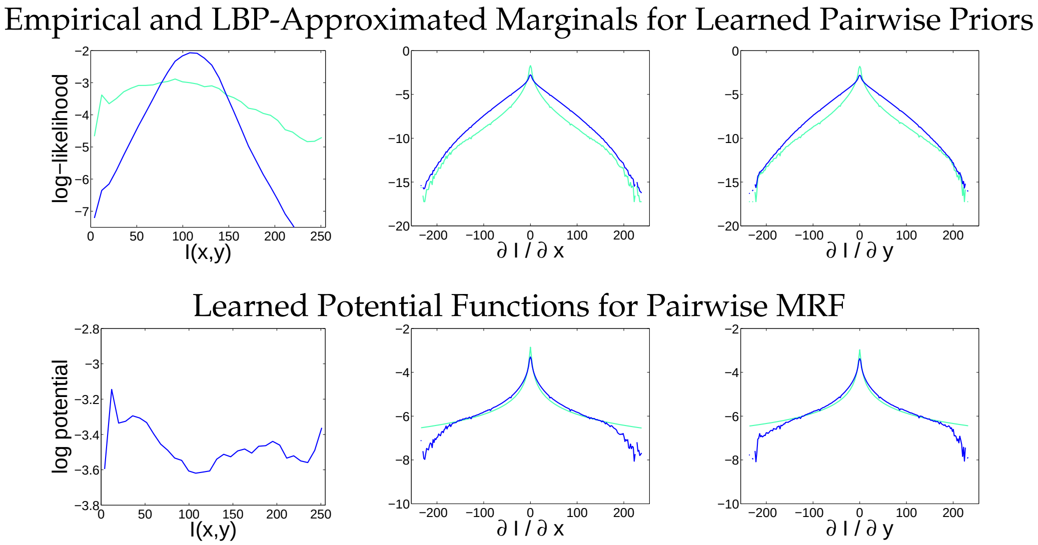


$\frac{\partial I}{\partial x}$
$\frac{\partial I}{\partial y}$
I

Figure 7.1: Factor graph used for learning pairwise MRF spatial priors for image denoising. Each circle represents a variable node (here, a pixel intensity), and each square represents a factor node for encouraging smoothness. The black outline denotes that the factor graph is connected as on a 2×2 torus. For a given set of potential functions, this factor graph monitors the estimated marginals for three linear image features: pixel intensity, horizontal derivative ($\partial I/\partial x$), and vertical derivative ($\partial I/\partial y$). Marginals for pixel intensity are equal to the beliefs at one of the variable nodes (here, we have chosen the upper-left node). Marginals for the pixel derivative values are estimated using two additional variable nodes, each of which is connected to a factor node of cliquesize three, whose messages are computed using linear constraint node techniques.

for learning parameters in ME models [60]. However, L-BFGS produced comparatively poor results for our application. One major difference between gradient descent methods and GIS for ME parameter learning is that GIS minimizes error in the log-marginals, rather than the marginals themselves. The histograms of natural image features are typically highly kurtotic, and histogram bins in the tails are often many orders of magnitude less likely than near-zero values. By minimizing error in the log-domain, GIS was able to capture these highly unlikely features far more accurately than gradient descent methods.

We will begin by learning the parameters of a pairwise-connected MRF



Empirical and LBP-Approximated Marginals for Learned Pairwise Priors

Learned Potential Functions for Pairwise MRF

Figure 7.2: Marginals and potential functions for the learned pairwise-connected MRF for natural images. The top row shows the empirical marginals (light green) and the LBP-approximated marginals, or beliefs (dark blue). The three linear features are the pixelwise image intensity, the horizontal derivative of intensity, and the vertical derivative. The bottom row shows the learned potential functions (dark blue). Each derivative feature is fit with a Student-t distribution (light green). The denoising results for the learned potential functions, and also for the fitted Student-t distributions, are listed in table 7.1.

of the form

$$p(I) = \prod_{x,y} f_p(I(x,y)) f_h(I(x,y) - I(x+1,y)) f_v(I(x,y) - I(x,y+1)) \quad (7.17)$$

Potential functions were represented as discrete histograms; 32 bins for f_p (to match the number of bins used during inference), and 101 bins for f_h and f_v (chosen arbitrarily). This MRF can be seen as a minimax entropy model with three linear features: a horizontal derivative, a vertical derivative, and the delta function (i.e. a prior over the intensity at a single pixel). For each of these linear features, beliefs (i.e. LBP-approximated marginals) are monitored by performing belief propagation on the torroidally connected factor graph in figure 7.1). For each of the two derivative features, an additional variable node was inserted into the pairwise-connected MRF to measure the beliefs for those linear features. A hard linear constraint node of clique-size 3 was used to enforce the linear relationship between these “monitor” nodes

	MAP		MMSE	
	$\sigma = 10$	$\sigma = 20$	$\sigma = 10$	$\sigma = 20$
Noisy Input Images	28.13	22.11	28.13	22.11
Hand-tuned Pairwise MRF using belief propagation [51]	30.73	26.66	NA	NA
2×2 FoE using gradient descent (algorithm from [73])	30.59	26.09	NA	NA
2×2 FoE using belief propagation (from [51])	30.89	27.29	NA	NA
2×2 FoE using LCNs, Fixed Histograms	31.41	27.12	31.51	27.29
2×2 FoE using LCNs & Adaptive Histograms	31.55	27.25	31.62	27.40
2×2 FoE using Particle/Histogram Hybrids	31.72	27.52	31.79	27.66
Pairwise MRF, GIS Learned Parameters	31.85	28.03	31.81	27.69
Higher-Order MRF, GIS Learned Parameters	32.17	28.07	32.10	28.03
Pairwise MRF, Learned Parameters fit with Student-t	32.17	28.07	32.10	28.03
Higher-Order MRF, Learned Parameters fit with Student-t	32.19	28.20	32.25	27.98
Full 5×5 FoE using gradient descent (algorithm from [73])	32.57	28.51	NA	NA

Table 7.1: Peak signal-to-noise ratio (PSNR), in decibels, for previous denoising models as well as pairwise and higher-order models learned using the efficient belief propagation techniques of section 4.1 combined with GIS. Each value gives the PSNR averaged over the ten images from the Berkeley segmentation database [61] used in [51]. PSNR is defined in equation 5.4. For each belief propagation algorithm, a MAP point estimate is approximated by choosing the maximal value of each marginal, and a MMSE point estimate is taken by computing the mean of each marginal. All belief propagation results were taken after 15 outerloop iterations (typically roughly 35 innerloop iterations).

and their neighboring pixel nodes. The addition of these additional nodes do not affect the beliefs computed by convergent belief propagation.

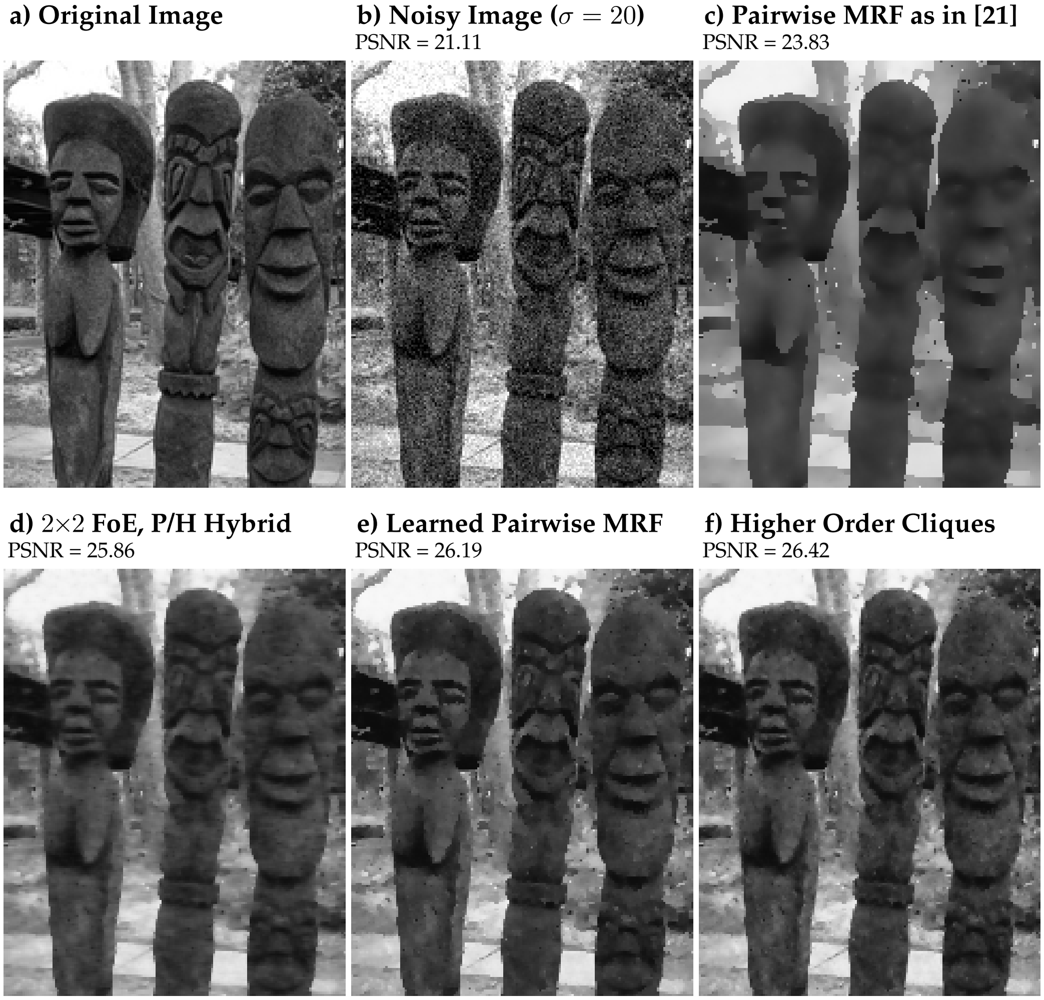
For each iteration of GIS, LBP was allowed to run for 50 outerloops of 5 innerloops each. This was more than enough to ensure convergence. GIS required under 30 iterations (evaluations of the beliefs $b_i^t(x_i)$) to converge, taking under 20 seconds total on a 3GHz Xeon.

The learned potential functions and the resulting LBP-approximated marginals are shown in figure 7.2. Next, to demonstrate the effectiveness of this learned spatial prior, image denoising was performed using this trained pairwise MRF. Again, while image denoising serves as a useful test case to evaluate image priors, the true power of this spatial prior is its ability to be exploited for more complex, ambiguous visual tasks that require belief propagation for good performance, such as shape-from-shading (Chapter 6), stereo [82], photometric stereo [84], image-based rendering [99], segmentation and matting [93]. Denoising results are given in table 7.1. Previous attempts at image denoising using pairwise-connected MRFs have resulted in blocky, piecewise-constant image regions (see figure 7.3**c**). Using parameters learned via GIS and belief propagation, pairwise-connected MRFs not only overcome this limitation, but they also outperform the 2 × 2 FoE model using belief propagation described in Chapter 5.

This is an important finding, because pairwise-connected MRFs remain highly popular as spatial priors in a variety of computer vision applications. Even using the efficient methods of belief propagation mentioned earlier, pairwise-connected MRFs are considerably faster than models with larger cliques. For a 240 × 160 image, each iteration of belief propagation required 38 seconds for the pairwise MRF, versus 57 seconds for 2 × 2 FoE. Also, belief propagation in pairwise MRFs tends to converge in fewer iterations. This can be a great advantage when performing inference over MRFs with more complex likelihood functions.

Next, we demonstrate our learning methods for MRFs with higher-order, non-pairwise cliques. Based on observations that derivative filters make effective features for spatial priors [112, 85], we add second order derivative features to our pairwise model. One way to do this by adding two additional variable nodes per pixel: P nodes that represent horizontal derivatives of image intensity, and Q nodes representing vertical derivatives. Hard linear constraint nodes (see section 4.1.3) of clique-size 3 are used to enforce the linear dependencies between P and Q nodes and pixel nodes. The prior probability of an image under this model is $\prod f_p(p_s)f_q(q_s)f_i(p_s - p_t)f_j(q_s - q_t)$ for neighboring pairs of P and Q nodes. Thus, neighboring P and Q nodes are connected within the MRF. Note that this overcomplete MRF matches the one used for shape-from-shading in Chapter 6, and thus may prove useful in improving 3D surface estimates. Performing image denoising using these learned parameters resulted in further improvement over the trained pairwise model. Results are listed in table 7.1.

In the introduction, we described three advantages of the learning methods used in this chapter: improved speed due to the use of linear constraint nodes, the ability to learn arbitrary potential functions instead of assuming that potential functions obey a Student-t distribution, and the ability to learn potential functions that compensate for the approximation that is inherent to loopy belief propagation. It is natural to ask which of these last two advantages explains the improved performance (despite smaller clique sizes) of the trained MRFs described above. To answer this question, we fit each learned potential function (other than the potential over single-pixel intensity, $f_p(I(x, y))$ with a Student-t distribution. For the learned pairwise MRF, those fits are shown in figure 7.2. We then repeated the image denoising experiments using these Student-t potential functions. The result



a) Original Image

b) Noisy Image ($\sigma = 20$)
PSNR = 21.11

c) Pairwise MRF as in [21]
PSNR = 23.83

d) $2{\times}2$ **FoE, P/H Hybrid**
PSNR = 25.86

e) Learned Pairwise MRF
PSNR = 26.19

f) Higher Order Cliques
PSNR = 26.42

Figure 7.3: MRF spatial priors applied to image denoising. Note that these priors are designed to be incorporated into a variety of other visual inference applications. **a)** The original image (from [61]). **b)** The original image with additive Gaussian noise of $\sigma = 20$. **c)** The output of belief propagation over a hand-designed pairwise-connected Markov Random Field similar to the model described in [21]. **d)** Denoising using three 2×2 Fields of Experts, using the Particle/Histogram message representation. **e)** Pairwise MRF with parameters learned using GIS and fit with Student-t distributions. **f)** Higher-Order MRF with parameters learned using GIS and fit with Student-t distributions.

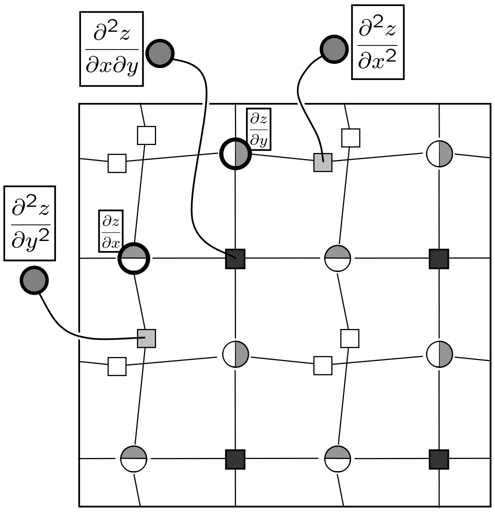

$\frac{\partial^2 z}{\partial x \partial y}$
$\frac{\partial^2 z}{\partial x^2}$
$\frac{\partial z}{\partial y}$
$\frac{\partial^2 z}{\partial y^2}$
$\frac{\partial z}{\partial x}$

Figure 7.4: Factor graph used for learning spatial priors for 3D surface shape. Similar to the SFS factor graph in figure 5.1, this graph includes variable nodes for the horizontal and vertical derivatives of depth at each pixel, and the linear dependencies among these values are enforced using hard linear constraint nodes of cliquesize four. Marginals for horizontal and vertical derivatives of depth are equal to the beliefs at one of the appropriate variable nodes (our choices shown here in heavy outline). Marginals for the three second order derivatives are monitored using a set of additional variable nodes. Messages to and from these nodes are made efficient using the linear constraint node technique. Note that, like in figure 7.1, the square outline surrounding the graph denotes that the nodes are connected as on a torus.

was a significant improvement. Numerical results are given in table 7.1, and example denoised images are shown in figure 7.3 **e** and **f**.

This improvement in quality shows that the good performance of the trained MRFs in this chapter was most likely due to the ability of GIS with LBP-approximated marginals to compensate for the approximation that is inherent to loopy belief propagation. It also suggests that, in this case, the ability to learn arbitrary potential functions for each linear image feature permitted too many parameters, and allowed the learning procedure to overfit the data. In other applications with more complex empirical marginals, however, the ability to learn arbitrary, non-parametric potential functions may outweigh the cost of learning many parameters.

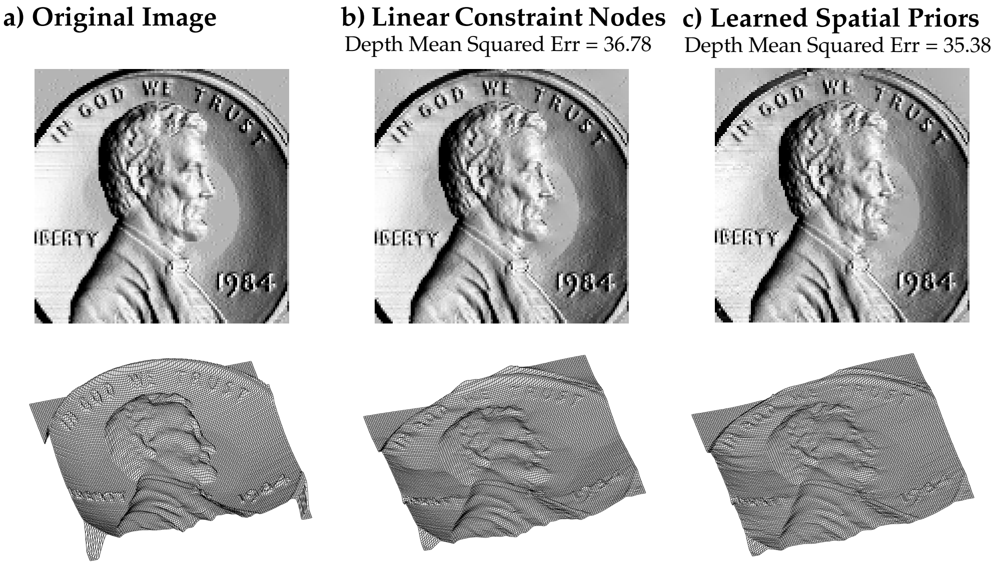
7.5 Results for Shape From Shading

As described in Chapter 6, the problem of Shape from Shading is highly underconstrained. Even under known lighting conditions and known surface reflectance properties, any given 2D image is completely consistent with a large number of possible 3D surfaces. In order to choose the most likely of these, we must first form an accurate model of what 3D shapes are most common in natural scenes. We must be able to train and exploit a strong spatial prior. This situation is the same for all depth inference scenarios. Stereo, photometric stereo, shape from texture and other depth cues all require accurate spatial priors to

In this section, I use GIS, combined with efficient belief propagation using linear constraint nodes, to learn a spatial prior model for natural 3D surface shapes. For the sake of comparison, I will learn potential functions for the same five linear features that were used in chapter 6. Recall that in chapter 6, the spatial prior utilized hand-tuned Laplace distributions over five linear features: the horizontal and vertical first derivatives $\partial z/\partial x$ and $\partial z/\partial y$, and also the three second-order derivatives $\partial^2 z/\partial x^2$, $\partial^2 z/\partial x \partial y$, and $\partial^2 z/\partial y^2$. In this section, I will use GIS to learn potential functions over these same five linear features.

To compute empirical marginals for each of these features, I used a suite of 28 images that was previously studied before in [68]. These scenes were chosen to each contain one a single object or surface type, such as statutes, building facades, rocky terrain, and foliage. This environment more closely matches the test images used for SFS, such as the penny image used in Chapter 6.

The results of the Shape from Shading algorithm using learned potentials are shown in figure 7.5. These learned priors reduce the mean squared error of the reconstructed surface slightly, from 36.78 to 35.38. While this constitutes some improvement, there is still a significant difference between the estimated shape and the ground-truth. This is especially true in the lower spatial frequencies of depth, where shading information is less informative, and small variations in local surface normals can accumulate to form gradual sloping deviations. It should be noted that even using hand-tuned Laplace priors, the SFS algorithm in figure 7.5b is successful in producing a reconstructed 3D shape with fairly natural smoothness properties, and so we cannot expect to see very large improvements. Our own acute perceptions of 3D shape when viewing the 2D penny image is at least partially a result of direct



a) Original Image

b) Linear Constraint Nodes
Depth Mean Squared Err = 36.78

c) Learned Spatial Priors
Depth Mean Squared Err = 35.38

Figure 7.5: **a)** The original 3D surface [109]. The rendering in this column serves as the input to the SFS algorithms in the next two columns. **b)** The surface recovered using the linear constraint node approach, as in Chapter 6 (figure 6.2). Recall that hand-tuned Laplace distribution potential functions were used as a spatial prior for this result. **c)** The surface recovered using the same linear constraint node technique as **b**, except using spatial priors learned via GIS. The errors listed here give the mean squared error of the final 3D depth reconstruction.

experience with its subject matter. This level of accuracy would be difficult to capture with a local 3D spatial prior. Without personal experience of the penny's 3D shape, it could be argued that the sharp depth discontinuity at the upper-right border of the penny is less likely, *a priori*, that the smooth reconstruction computed by the algorithm in figure 7.5c.

7.6 Conclusions

In this chapter, I have shown that learning using GIS with belief propagation offers a significant speed advantage over sampling methods for learning the potentials functions of maximum entropy graphical models. This advantage is made possible by the ability of belief propagation to compute not only MAP or MMSE point estimates of a probability distribution, but also entire marginals over single variables. The ability to compute marginals is a seldommentioned advantage that belief propagation has over other sophisticated inference techniques, such as graph cuts. This ability is a great advantage when it comes to learning parameters.

Training potential functions using GIS is fast enough that repeated applications may allow the image features $\vec{v}_i$ to be learned as well, either by feature selection from a set of predetermined candidates, or by following gradient descent. Efficient parameter learning may also benefit the training of conditional random fields [50], where a separate set of potential functions must be trained for different values of the input.

Another major advantage of the learning methods described here is that they allow us to learn parameters of a MRF that *compensate* for the approximation that is inherent to the use of belief propagation. Belief propagation finds a set of "beliefs" $b_i(x_i)$ for a MRF $P(\vec{X}|\theta)$ that approximate the true marginals $p_i(x_i)$ of P. Most learning procedures work by searching for parameters θ that minimize the KL-divergence between the MRF distribution $P(\vec{X}|\theta)$ and some empirical distribution $P_0(\vec{X})$. For learning potential functions over linear features, it can be shown that this is equivalent to finding potential functions such that the true marginals $p_i(x_i)$ of $P(\vec{X}|\theta)$ match the true marginals $\hat{o}_i(x_i)$ of the empirical distribution $P_0(\vec{X})$. Because belief propagation computes only approximate marginals, this means that the marginals $b_i(x_i)$ that are the result of belief propagation inference may not match the empirical marginals $\hat{o}_i(x_i)$. To minimize the error of belief propagation, a better strategy is to minimize the divergence between the empirical marginals $\hat{o}_i(x_i)$ and the *approximate* marginals $b_i(x_i)$ that are the result of belief propagation. This is the strategy employed by the learning methods described here.

When belief propagation is applied to MRFs trained using the methods of this chapter, the result is a significant improvement over previous training methods. Using this learning approach, even a simple pairwise MRF, can be trained to outperform the 2×2 FoE model trained using Contrastive Divergence [31]. This is an important finding, because pairwise MRFs are far simpler to implement than higher-order cliques, and also significantly more efficient in terms of speed and memory. In the example presented here, the pairwise MRF was nearly twice as fast as the 2×2 FoE, even using the efficient methods of Chapter 4. Previous to the results presented here, pairwise MRFs were thought to be highly limited as a model for spatial priors, capable only of results similar to those of figure 7.3c. The discovery that pairwise MRFs can produce effective spatial priors could prove important for efficient visual inference for ambiguous problems, such as stereo or novel scene synthesis.

It is important to note that pairwise MRFs have significantly fewer parameters than the 2×2 FoE model. Pairwise MRFs have only two cliques per pixel, each of size two, whereas the 2×2 FoE model has three cliques per pixel, each of size four. In fact, the pairwise MRF can be seen as a submodel of the 2×2 FoE: for some choice of linear features and potentials, the 2×2 FoE can be made to emulate any pairwise MRF. It is significant, then, that pairwise MRFs trained using our method outperform the richer 2×2 FoE model trained using Contrastive Divergence.

Another contribution of this chapter is the ability to use GIS with belief propagation to efficiently learn parameters for higher-order cliques. These higher-order models are better able to capture the rich statistical structure present in natural images and 3D scenes. We demonstrate this by using higher-order MRFs, trained using GIS with LBP, for image denoising, and show an improvement over the trained pairwise models. In fact, the denoising results produced by these higher-order models come surprisingly close to the specialized state-of-the-art methods of image denoising [66, 73] which cannot be generalized to be used as spatial priors for more complex inference tasks such as stereo or shape-from-shading. For some images in the testing suite, belief propagation in the trained higher-order MRFs actually surpassed the performance of these state-of-the-art methods. The ability to exploit a spatial prior of this caliber for stereo, shape-from-shading, novel scene synthesis, super-resolution, segmentation, and matting may yield significant improvements in each of these applications.

Chapter 8

Conclusions

The problem of inferring underlying scene properties such as 3D shape from images is both complex and underconstrained. To solve these problems effectively, sophisticated statistical inference techniques must be developed to simultaneously resolve ambiguity and handle complex statistical relationships. This thesis proposes a mathematical framework for statistical inference that can efficiently handle ambiguous yet complex higher-order statistical relationships, making it well suited to handle difficult visual inference tasks. I then apply this methodology to three central issues related to the depth inference: the inference of shape from shading, the use of strong spatial priors, and the ability to train statistical models from empirical data.

These three tasks were chosen to be representative of the issues faced by depth inference problems in general, as well as other visual inference tasks. In order to infer 3D shape or other scene properties from an image, it is necessary to exploit visual cues, to resolve ambiguity by using strong spatial priors, and preferably, to be able to learn both priors and cues from real scenes. The applications presented in this thesis are representative of these goals.

Some of the applications presented in this thesis were also chosen as particularly difficult example problems, to demonstrate the ability of the approach to scale to more general inference problems. For example, one potential weakness of belief propagation is that the Bethe approximation is expected to deteriorate for MRFs with many tight loops and for cliques with high-energy deterministic or nearly-deterministic potentials [28]. The shape-from-shading application in Chapter 6 faced both of these problems. Other depth inference cues, such as stereo, occlusion contours, texture, perspective, or shadow, are not nearly so deterministic. In fact, for more natural scenes, which include uncertain surface reflectances and lighting conditions, shading cues will also become less deterministic than the Lambertian constraint of the classic problem definition of shape-from-shading. Belief propagation should perform better for these lower-energy cues. The fact that belief propagation performed so well for classic SFS is suggestive that belief propagation with linear constraint nodes should provide an adequate framework for more realistic depth inference scenarios.

Shape from shading also provides a good test of our depth inference framework because it is known to be a very difficult inference problem. The problem of SFS has been studied since the 1920s, when astronomers sought to understand the surface of the moon [35]. Since then, the problem has received considerable attention from the field of computer vision. Despite years of research, previous state-of-the-art methods for solving SFS are still regarded as unsatisfactory. In a 1999 survey, Zhang et. al. [109] conclude that:

> "All the SFS algorithms produce generally poor results when given synthetic data ..."

In contrast, the LBP with linear constraint nodes approach of Chapter 6 is quite successful at inferring a plausible 3D surface to match the input image. In particular, the surface inferred via belief propagation satisfies both the Lambertian and the integrability constraints nearly perfectly, meaning that when rendered under illumination conditions identical to the input surface, the resulting image matches the input image almost exactly. For previous SFS approaches, the rendering of the inferred surface only barely resembles the coarse features of the input image.

The ability of belief propagation with linear constraint nodes to solve SFS effectively is one of the central points of this thesis. However, perhaps even more important is the ability of this approach to generalize to include more depth cues, to exploit stronger spatial priors, and to scale to handle scenes and situations considerably more general than those demanded by the strict requirements of the original problem definition of SFS.

In the time since the 1999 survey by Zhang et. al. [109], the study of SFS has primarily focused on special cases that reduce the ambiguity of the problem by further restricting the subclass of scenes that are eligible for analysis [71]. Algorithms have been developed to solve SFS as long as the single-point illumination source occupies the same point in space as the camera [70]. This constraint is in addition to the standard constraints already imposed by classical SFS. Other SFS methods require that no point in the image lie in attached shadow, which means that no surface normal is greater than 90 degrees from the illumination angle [71].

The viewpoint of this thesis is that further constraining the SFS problem is moving in the wrong direction. The requirements imposed by classical SFS on which scenes are eligible for analysis is already restrictive enough to make real-world applications difficult to come by. The ambiguity of the SFS problem cannot be ignored forever. Instead of restricting the problem domain until SFS is unambiguous or “well-posed”, real progress on SFS will require developing methods that allow reasoning under uncertainty; to use modern statistical inference techniques that can simultaneously handle complex mathematical relationships as well as underconstrained, ambiguous circumstances. It is more important to work to generalize SFS in ways that allow us to *relax* the stiff restrictions imposed by the classical formulation of SFS than to seek out ways of restricting it further. The approach to SFS presented in this thesis not only achieves ground-breaking performance on classical SFS problems, but it also promises to generalize to considerably more flexible depth inference scenarios. The SFS approach presented in this thesis can be generalized in straightforward ways to handle non-Lambertian surface reflectance, to work in lighting conditions other than single point-source lighting, such as multiple light sources or diffuse lighting, to exploit multiple depth cues, and to exploit strong spatial priors

The remaining two applications presented in this thesis, the use of higher-order spatial priors and learning, are both designed to facilitate generalizing SFS and also to improve other depth-inference and scene property inference problems, such as stereo, photometric stereo, novel scene synthesis, segmentation, and matting. All of these visual inference problems are highly ambiguous in nature, and each of them requires selecting the most probable configuration from large volumes of plausible choices. Resolving such ambiguity requires exploiting a strong prior. In the context of SFS, recall that for any given 2D input image, there can be huge number of possible 3D surfaces which all render, under lighting conditions identical to the input scene, to match the input image exactly. In fact, the dimensionality of the space of surfaces that are consistent with the input image can be as large as $W + H + S$, where W and H are the image width and height, respectively, and S is the number of pixels that lie in attached shadow. The only method for choosing one 3D surface from this high-dimensional space is to exploit a strong spatial prior, $P(Z)$, that identifies which 3D surfaces are likely to exist in real scenes. Recall that the SFS algorithm in Chapter 6 was able to find a 3D surface that matched the input image nearly exactly, meaning that the computed 3D surface lies in that $(W + H + S)-$dimensional space of surfaces consistent with the input image. The only way to improve the output of this algorithm is to improve the spatial prior $P(Z)$ exploited by the algorithm.

Exploiting strong, higher-order spatial priors is precisely the subject of Chapter 5. Higher-order spatial priors have been developed in connection with image denoising [66, 73] in the past. However, exploiting such strong priors from within the statistical inference framework of belief propagation has historically been problematic due to the exponential running time of computing belief propagation messages through large MRF cliques. Chapter 5 shows how these problems can be overcome using the linear constraint node technique introduced in Chapter 4. The ability to exploit strong spatial priors without relying on weaker inference methods such as gradient descent (as used by Roth et. al. to perform denoising [73]) may be of significant benefit to a variety of visual inference tasks such as SFS, stereo, novel scene synthesis, segmentation, and matting.

The issue of strong spatial priors was explored further in Chapter 7, which described how belief propagation and the computational shortcuts of Chapter 4 could be used not only for inference, but also for learning the parameters of graphical models from empirical data. The learning methods of Chapter 7 allow spatial priors to be learned efficiently from natural scenes. Furthermore, by minimizing the error of the output of belief propagation, rather than minimizing the error of the idealized MRF, MRFs trained using the methods of Chapter 7 can compensate for the approximation that is inherent to the belief propagation method of inference. This presents a significant advantage over other learning methods, and the spatial priors trained using these methods significantly outperform models trained using previous state of the art methods such as contrastive divergence [31]. Remarkably, even a simple pairwise-connected MRF, when trained using the methods of Chapter 7, can outperform considerably more sophisticated spatial prior models such as 2×2 Fields of Experts that were trained using contrastive divergence. We show how strong spatial priors trained using these techniques produce a significant advantage for image denoising and also for shape from shading. Again, these strong spatial priors can also extend to several other visual inference problems such as stereo novel scene synthesis, segmentation, and matting.

Ultimately, the ability to train MRFs efficiently will be useful not only for learning strong spatial priors, but also for learning data likelihood from natural scenes. As described in the introduction, little is known about the statistics of natural 3D scenes. Many previous approaches to the inference of 3D shape have been based on physical models of image formation that rely on untested parameters, assumptions, and oversimplifications. Realistic depth inference for natural images will require studying 3D shapes in real 3D scenes, measuring what depth cues really exist in nature, measuring their relative strength, and estimating their parameters and exact forms from real scenes. The technical capability to study the joint statistics of natural images and their underlying 3D shapes is only now becoming feasible. Using laser-acquired range images with coregistered color images such as our database (described in section 2.3 and used in section 7.5), we will be able to measure from real 3D scenes what statistical trends actually exist in natural environments. Statistical learning techniques like those presented in Chapter 7 will then help to exploit these empirical trends from the probabilistic framework of graphical models. Then, using the methods outlined in this thesis, we can approach the problem of inferring 3D structure in general natural scenes.

8.1 Future Work

In the short term, I will continue to work on statistical methods for the inference of 3D shape. I will develop ways to extend my existing techniques to unrestricted natural images. This includes exploiting stronger spatial priors and integrating additional monocular depth cues, such as occlusion, shadow, texture, and perspective. The inference of depth from monocular cues is a very difficult problem, one which I expect to be a subject of active ongoing research for several decades.

I also intend to continue work to develop the machine learning tools that can best achieve these aims. The efficient approaches to statistical inference and parameter learning I have developed can be extended in several ways to tackle harder problems more efficiently. Additionally, the machine learning techniques I have developed in the process of completing this thesis have promising applications to several computer vision problems, such as super-resolution, novel scene synthesis, stereo, segmentation, and matting. I hope to expand my research in all of these areas.

I will also continue to explore the statistical regularities of natural scenes.

The study of the joint statistics of natural scenes is a wide open field, and one that promises to yield insights into both computer vision and visual neuroscience. Future projects include the study of occlusion contours, and how their stochastic geometry changes over scale. A better understanding of occlusion will help to develop algorithms that can be applied to cluttered scenes with multiple 3D surfaces.

Finally, I will continue to explore how statistical inference can be achieved in the brain. Already, for much of the work described in this thesis, I am involved in analogous projects aimed at exploring how visual inference occurs in the brain. Just as I explore the use of spatial priors in Chapters 5 and 7, I am also working to understand how spatial priors are used the brain to disambiguate shape-from-stereo in visual area V1 [75]. Just as I consider ways in which multiple depth cues can be integrated in Chapter 6, I also work to understand how stereo and shadow cues are combined in the brain, by recording and analyzing the response of V1 neurons to stereo and shadow cues [69]. Ongoing and exciting developments in machine learning techniques, as well as new methods for understanding neurological function, are opening up rich new avenues for interdisciplinary research, and I am fortunate to be in a position to explore these new areas.

Bibliography

[1] Kenichi Arakawa and Eric Krotkov. Fractal modeling of natural terrain: Analysis and surface reconstruction with range data. *Graphical Models and Image Processing*, 58(5):413–436, September 1996.

[2] M. Ashley. Concerning the significance of light in visual estimates of depth. *Psychological Review*, 5(6):595–615, 1898.

[3] Joseph J. Atick, Paul A. Griffin, and A. Norman Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8(6):1321–1340, 1996.

[4] Anthony J. Bell and Terrence J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 37:2228–3327, 1997.

[5] J. Ben-Arie and D. Nandy. A Neural Network Approach for Reconstructing Surface Shape from Shading. In *Proceedings of the IEEE International Conference on Image Processing (volume II)*, pages 972–976, Chicago, Illinois, USA, October 1998.

[6] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 237–252, London, UK, 1992. Springer-Verlag.

[7] Hans L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–21, 1993.

[8] H. Carr. *An Introduction to Space Perception*. Longmans, Green and Co, New York, 1935.

[9] Hui Cheng and Charles A. Bouman. Multiscale bayesian segmentation using a trainable context model. *IEEE Transactions on Image Processing*, 10(4):511–525, 2001.

[10] James Coughlan and Huiying Shen. Shape matching with belief propagation: Using dynamic quantization to accomodate occlusion and clutter. In *CVPRW*, page 180, 2004.

[11] James M. Coughlan and Alan L. Yuille. Algorithms from statistical physics for generative models of images. *Image and Vision Computing*, 21(1):29–36, 2003.

[12] James M. Coughlan and Alan L. Yuille. Algorithms from statistical physics for generative models of images. *Image Vision Comput.*, 21(1):29–36, 2003.

[13] J. Coules. Effect of photometric brightness on judgments of distance. *Journal of Experimental Psychology*, 50:19–25, 1955.

[14] J. E. Cryer, P. S. Tsai, and M. Shah. Integration of shape from shading and stereo. *Pattern Recognition*, 28(7):1033–1043, July 1995.

[15] J. E. Cutting and P. M. Vishton. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In William Epstein and Sheena J Rogers, editors, *Perception of space and motion*, Handbook of perception and cognition, pages 69–117. Academic Press, San Diego, CA, USA, 1995.

[16] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Math. Statistics*, 43:1470–1480, 1972.

[17] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems.* The MIT Press, December 2001.

[18] Stanley R. Deans. *The Radon Transform and Some of Its Applications.* John Wiley & Sons, 1983.

[19] J.-D. Durou, M. Falcone, and M. Sagona. A Survey of Numerical Methods for Shape from Shading. Rapport de Recherche 2004-2-R, Institut de Recherche en Informatique de Toulouse, Toulouse, France, January 2004.

[20] M. Farne. Brightness as an indicator to distance: Relative brightness per se or contrast with the background? *Perception*, 6:287–293, 1977.

[21] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, volume 1, pages 261–268, 2004.

[22] David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of Optical Society of America*, 4(12):2379–2394, 1987.

[23] William T. Freeman, Egon Pasztor, and Owen T. Carmichael. Learning low-level vision. *Int. J. Comp. Vis.*, 40(1):25–47, 2000.

[24] Jerome H. Friedman, Werner Stuetzle, and Anne Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

[25] J. J. Gibson. *The Ecological Approach to Visual Perception.* Mifflin, Boston, 1979.

[26] Feng Han and Song Chun Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *IEEE 1st International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK)*, pages 12–20. IEEE Computer Society, 2003.

[27] Tom Heskes. Stable fixed points of loopy belief propagation are local minima of the Bethe free energy. In *NIPS*, pages 343–350, 2002.

[28] Tom Heskes. On the uniqueness of loopy belief propagation fixed points. *Neural Comp.*, 16(11):2379–2413, 2004.

[29] Tom Heskes, Kees Albers, and Bert Kappen. Approximate inference and constrained optimization. In *UAI*, pages 313–320, 2003.

[30] Geoffrey Hinton. Products of experts. In *International Conference on Artificial Neural Networks*, volume 1, pages 1–6, 1999.

[31] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

[32] D. Hoiem, A. Stein, A.A. Efros, , and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, October 2007.

[33] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH*, August 2005.

[34] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *International Conference of Computer Vision (ICCV)*. IEEE, October 2005.

[35] Berthold K. P. Horn. Obtaining shape from shading information. pages 123–171, 1989.

[36] C. Q. Howe and D. Purves. Range image statistics can explain the anomalous perception of length. *Proc. Nat. Acad. Sci.*, 99:13184–13188, 2002.

[37] Jinggang Huang, Ann B. Lee, and David Mumford. Statistics of range images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1324–1331, 2000.

[38] Jinggang Huang and David Mumford. Statistics of natural images and models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 541–547, 1999.

[39] Michael Isard. Pampas: Real-valued graphical models for comnputer vision. In *CVPR*, pages 613–620, 2003.

[40] A. Jalobeanu, F.O. Kuehnel, and J.C. Stutz. Modeling images of natural 3d surfaces: Overview and potential applications. In *Proc. of IEEE conf. on Computer Vision and Pattern Recognition, Graphical Model-Based Vision workshop*, Washington DC, USA, Jul 2004.

[41] T. Z. Jiang, B. Liu, Y. L. Yu, and D. J. Evans. A Neural Network Approach to Shape from Shading. *International Journal of Computer Mathematics*, 80(4):433–439, April 2003.

[42] D.C. Knill and D. Kersten. Learning a near-optimal estimator for surface shape from shading. *Computer Vision, Graphics and Image Processing*, 50(1):75–100, April 1990.

[43] Daphne Koller, Uri Lerner, and Dragomir Anguelov. A general algorithm for approximate inference and its application to hybrid bayes net. In *UAI*, pages 324–33, 1999.

[44] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1568–1583, 2006.

[45] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 508–515. IEEE, 2001.

[46] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 65–81, London, UK, 2002. Springer-Verlag.

[47] Alexander Kozlov and Daphne Koller. Nonuniform dynamic discretization in hybrid networks. In *UAI*, pages 314–32, 1997.

[48] Frank R. Kschischang and Brendan J. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal of Selected Areas in Communications*, 16(2):219–230, 1998.

[49] Sanjiv Kumar and Martial Hebert. Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–202, 2006.

[50] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML: International Conference on Machine Learning*, 2001.

[51] Xiangyang Lan, Stefan Roth, Daniel P. Huttenlocher, and Michael J. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV*, pages 269–282, 2006.

[52] M. S. Langer and S. W. Zucker. Shape from Shading on a Cloudy Day. *Journal of the Optical Society of America - Part A: Optics, Image Science, and Vision*, 11(2):467–478, February 1994.

[53] M.S. Langer and H.H. Blthoff. Perception of shape from shading on a cloudy day. Technical Report 73, Tbingen, Germany, oct 1999.

[54] Ann B. Lee, David Mumford, and Jinggang Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *Int. J. Comput. Vision*, 41(1-2):35–59, 2001.

[55] K.M. Lee and C.C.J. Kuo. Shape from shading with a linear triangular element surface model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(8):815–822, 1993.

[56] Tai Sing Lee and David Mumford. Hierarchical bayesian inference in the visual cortex. *J. Opt. Soc. Amer. A*, 20:1434–1448, 2003.

[57] S. R. Lehky and T. J. Sejnowski. Network model for shape–from–shading: Neural function arises from both receptive and projective fields. *Nature*, 333:452–454, June 1988.

[58] Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 787–794. MIT Press, Cambridge, MA, 2006.

[59] S. Livens, P. Scheunders, G. Van de Wouwer, P. Vautrot, and D. Van Dyck. Wavelets for texture analysis, an overview. In *Proc. IEE International Conference on Image Processing and Applications*, pages 581–585, Dublin, July 1997.

[60] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *CoNLL-2002*, pages 49–55. Taipei, Taiwan, 2002.

[61] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, pages 416–423, 2001.

[62] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475, 1999.

[63] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[64] A. P. Pentland. Linear Shape From Shading. *International Journal of Computer Vision*, 4(2):153–162, March 1990.

[65] Nemanja Petrovic, Ira Cohen, Brendan J. Frey, Ralf Koetter, and Thomas S. Huang. Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In *CVPR*, pages 743–748, 2001.

[66] 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, 2003.

[67] Brian Potetz and Tai Sing Lee. Statistical correlations between two-dimensional images and three-dimensional structures in natural scenes. *J. Opt. Soc. Amer. A*, 20(7):1292–1303, 2003.

[68] Brian Potetz and Tai Sing Lee. Scaling laws in natural scenes and the inference of 3d shape. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1089–1096. MIT Press, Cambridge, MA, 2006.

[69] Brian Potetz, Jason Samonds, and Tai-Sing Lee. Disparity and luminance preferences are correlated in macaque V1, matching natural scene statistics. In *Society for Neuroscience*, Atlanta, GA, 2006.

[70] E. Prados and O. Faugeras. "perspective shape from shading" and viscosity solutions. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 826, Washington, DC, USA, 2003. IEEE Computer Society.

[71] Emmanuel Prados, Fabio Camilli, and Olivier Faugeras. A unifying and rigorous shape from shading method adapted to realistic data and applications. *Journal of Mathematical Imaging and Vision*, 25(3):307–328, 2006.

[72] V. S. Ramachandran. Perception of shape from shading. *Nature*, 331:163–166, 1988.

[73] Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005.

[74] D. L. Ruderman and W. Bialek. Statistics of natural images: scaling in the woods. *Physical Review Letters*, 73:814–817, 1994.

[75] Jason Samonds, Brian Potetz, and Tai-Sing Lee. Neurophysiological evidence of cooperative mechanisms for stereo computation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1201–1208. MIT Press, Cambridge, MA, 2007.

[76] Ashutosh Saxena, Sung H. Chung, and Andrew Ng. Learning depth from single monocular images. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.

[77] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

[78] Daniel Scharstein and Richard Szeliski. Middlebury stereo evaluation. `http://vision.middlebury.edu/stereo`, April 2008.

[79] J. Shan and S. D. Lee. Quality of building extraction from ikonos imagery. *Journal of Surveying Engineering*, 131(1):27–32, 2005.

[80] Solomon E. Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, August 1994.

[81] Erik Sudderth, Alexander Ihler, William Freeman, and Alan Willsky. Nonparametric belief propagation. In *CVPR*, 2003.

[82] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, 2003.

[83] R.T. Surdick, E.T. Davis, R.A. King, and L.F. Hodges. The perception of distance in simulated visual displays: A comparison of the effectiveness and accuracy of multiple depth cues across viewing distances. *Presence: Teleoperators and Virtual Environments*, 6:513–531, 1997.

[84] Kam Lun Tang, Chi Keung Tang, and Tien Tsin Wong. Dense photometric stereo using tensorial belief propagation. In *CVPR*, pages 132–139, 2005.

[85] Marshall F. Tappen. Utilizing variational optimization to learn markov random fields. In *CVPR 2007: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, Minneapolis, MN, USA, 2007.

[86] I. L. Taylor and F. C. Sumner. Actual brightness and distance of individual colors when their apparent distance is held constant. *The Journal of Psychology*, 19:79–85, 1945.

[87] Y. W. Teh and M. Welling. The unified propagation and scaling algorithm. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

[88] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1226–1238, 2002.

[89] Jose R. A. Torreão. Bayesian shape estimation: Shape-from-shading and photometric stereo revisited. *Machine Vision and Applications*, 8(3):163–172, May 1995.

[90] P.S. Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing*, 12(8):487–498, 1994.

[91] C. Tyler. Diffuse illumination as a default assumption for shape from shading in the absence of shadows. *The Journal of imaging science and technology*, 42(4):319–325, 1998.

[92] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. A new class of upper bounds on the log partition function. *Information Theory, IEEE Transactions on*, 51(7):2313–2335, July 2005.

[93] Jue Wang and Michael F. Cohen. An iterative optimization approach for unified image segmentation and matting. *ICCV*, pages 936–943, 2005.

[94] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[95] Yair Weiss and William T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001.

[96] Yair Weiss and William T. Freeman. What makes a good model of natural images? In *CVPR 2007: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, Minneapolis, MN, USA, 2007.

[97] M. Welling and Y. W. Teh. Belief optimization for binary networks: a stable alternative to loopy belief propagation. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, volume 17, 2001.

[98] Andrew Witkin, Demetri Terzopoulis, and Michael Kass. Signal matching through scale space. In *Readings in computer vision: issues, problems, principles, and paradigms*, pages 759–764. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[99] O. J. Woodford, I. D. Reid, P. H. S. Torr, and A. W. Fitzgibbon. Fields of experts for image-based rendering. In *Proceedings of the 17th British Machine Vision Conference, Edinburgh*, volume 3, pages 1109–1108, 2006.

[100] M. Wright and T. Ledgeway. Interaction between Luminance Gratings and Disparity Gratings. *Spatial Vision*, 17(1–2):51–74, 2004.

[101] Tai Pang Wu and Chi Keung Tang. Dense photometric stereo using a mirror sphere and graph cut. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 140–147, 2005.

[102] Zhiyong Yang and Dale Purves. Image/source statistics of surfaces in natural scenes. *Network: Computation in Neural Systems*, 13(3):371–390, 2003.

[103] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.

[104] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, pages 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[105] 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, 2005.

[106] Alan L. Yuille. An algorithm to minimize the Bethe free energy. In *EMMCVPR*, 2001.

[107] Alan L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.

[108] Nevin Lianwen Zhang and David Poole. A simple approach to bayesian network computations. In *Proc. of the Tenth Canadian Conference on Artificial Intelligence*, pages 171–178. 1994.

[109] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, 1999.

[110] Qinfen Zheng and Rama Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(7):680–702, 1991.

[111] Song Chun Zhu and David Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(11):1236–1250, 1997.

[112] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its applications to texture modeling. *Neural Computation*, 9:1627–1660, 1997.

[113] Song Chun Zhu, Ying Nian Wu, and David Mumford. Frame : Filters, random fields and maximum entropy — towards a unified theory for texture modeling. *Int'l Journal of Computer Vision*, 27(2):1–20, 1998.