Efficient Belief Propagation for Vision Using Linear Constraint Nodes

Brian Potetz
Carnegie Mellon University
5000 Forbes Ave
bpotetz@cs.cmu.edu

Abstract

Belief propagation over pairwise connected Markov Random Fields has become a widely used approach, and has been successfully applied to several important computer vision problems. However, pairwise interactions are often insufficient to capture the full statistics of the problem. Higher-order interactions are sometimes required. Unfortunately, the complexity of belief propagation is exponential in the size of the largest clique. In this paper, we introduce a new technique to compute belief propagation messages in time linear with respect to clique size for a large class of potential functions over real-valued variables.

We demonstrate this technique in two applications. First, we perform efficient inference in graphical models where the spatial prior of natural images is captured by 2×2 cliques. This approach shows significant improvement over the commonly used pairwise-connected models, and may benefit a variety of applications using belief propagation to infer images or range images. Finally, we apply these techniques to shape-from-shading and demonstrate significant improvement over previous methods, both in quality and in flexibility.

1. Introduction

In the past few decades, the application of probabilistic models for solving computer vision problems has lead to significant advances. Many of these probabilistic models can be simplified by factoring large probability distributions using graphical models. Recently, the method of loopy belief propagation has been shown to produce excellent results in several real-world computer vision problems [4, 19, 20, 10, 13]. However, this method has some drawbacks. The most serious is that the running time of belief propagation is exponential in the size of the largest graph clique. This means that for problems with many labels or real-valued variables, graphical representations are typically limited to pairwise interactions between variables. Unfortunately, for many problems in computer vision, pairwise interactions fail to capture the rich statistical distribution of the problem domain. For example, natural images exhibit rich higher-order statistics that cannot be captured by pairwise connected Markov Random Fields (MRFs). In section 3, we 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. In section 3.4, we further improve on the efficiency and performance of belief propagation by presenting a nonparametric, particle-like representation of belief propagation messages that is simultaneously compatible with higher-order non-pairwise interactions and also with recent extensions to belief propagation that guarantee convergence [6].

These advancements allow us to efficiently solve inference problems that were previously unavailable to belief propagation. In section 4, we show that a prior model of natural images using 2×2 MRF cliques strongly outperforms pairwise-connected models. The ability to use more accurate models of image or range image priors has the potential to significantly aid the performance of several computer vision applications, including stereo [19], photometric stereo [20], shape-from-shading (section 5) and image-based rendering. Finally, in section 5, we apply our methods to the problem of shape-from-shading. We show that our approach leads to significant advancements in performance over previous methods. Additionally, due to the flexible nature of belief propagation, rather than hardcoding a reflectance function into the algorithm, our approach can support not only arbitrary reflectance functions, but also uncertainty in reflectance, lighting direction, and surface albedo.

2. Belief Propagation

Belief propagation is a method for estimating the marginals of a multivariate probability distribution of the form:

$$p(\vec{X}) = \prod \phi_i(\vec{x_i}) \qquad \vec{x_i} \subset \vec{X} \tag{1}$$

Such probability distributions are often represented in the form of a *factor graph*. A factor graph is a bipartite graph in which each potential function $\phi_i(\vec{x_i})$ is represented by a factor node f, which is connected to one variable node v for each element of the vector $\vec{x}_i$. An example factor graph is depicted in figure 3. Sum-product belief propagation estimates the marginals $b(x_i) = \sum_{X \backslash x_i} p(\vec{x})$ by iteratively computing *messages* along each edge of the graph according to the equations:

$$m^t_{i \to f}(x_i) = \prod_{g \in \mathcal{N}(i) \backslash f} m^{t-1}_{g \to i}(x_i) \tag{2}$$

$$m^t_{f \to i}(x_i) = \sum_{\vec{x}_{\mathcal{N}(f) \backslash i}} \left(\phi_f\left(\vec{x}_{\mathcal{N}(f)}\right) \prod_{j \in \mathcal{N}(f) \backslash i} m^t_{j \to f}(x_j) \right) \tag{3}$$

$$b^t_i(x_i) \propto \prod_{g \in \mathcal{N}(i)} m^t_{g \to i}(x_i) \tag{4}$$

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 [6]. Here, $b_i(x_i)$ is the estimated marginal of variable i. 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. If the most likely value of $\vec{X}$ is desired, also known as the maximum a posteriori (MAP) point estimate, then the integrals of equation 3 are replaced by supremas. This is known as max-product belief propagation.

When the underlying factor graph is a tree, one iteration of sum-product belief propagation is guaranteed to compute the correct marginals [21]. When the factor graph contains loops, the messages must be updated iteratively until convergence is reached. This is known as *loopy belief propagation* (LBP), and convergence is no longer guaranteed.

3. Efficient Belief Propagation

Belief propagation has been applied successfully to a variety of computer vision problems [4, 19, 20]. However, in many computer vision problems, belief propagation is prohibitively slow. The high-dimensional summation in equation 3 has a complexity of $\mathcal{O}(M^N)$, where M is the number of possible labels for each variable, and N is the number of neighbors of f. In many computer vision problems, variables are continuous or have many labels. In these cases, applications of belief propagation have nearly always been restricted to pairwise connected Markov Random Fields, where each potential function in equation 1 depends on only two variable nodes [4, 19]. However, pairwise connected models are often insufficient to capture the full complexity of the joint distribution of the problem. In this section, we describe methods to efficiently compute belief propagation messages over continuous random variables for a wide range of higher-order (non-pairwise) potential functions.

3.1. Linear Constraint Nodes

Consider potential functions of the form

$$\phi(\vec{x}) = g(\vec{x} \cdot \vec{v}) \tag{5}$$

where $\vec{x}$ and $\vec{v}$ are vectors of length N. Normally, computing messages from this factor node 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:

$$M_1(x_1) = \iiint g(v_1x_1 + v_2x_2 + v_3x_3 + v_4x_4) \, m_2(x_2)m_3(x_3)m_4(x_4)\, dx_2\, dx_3\, dx_4 \tag{6}$$

$$= \iiint J\, g(v_1x_1 + y_1)\, m_2(\frac{y_1 - y_2}{v_2}) \, m_3(\frac{y_2 - y_3}{v_3})\, m_4(\frac{y_3}{v_4})\, dy_1\, dy_2\, dy_3 \tag{7}$$

$$\propto \int g(v_1x_1 + y_1)\Big(\int m_2(\frac{y_1 - y_2}{v_2}) \Big(\int m_3(\frac{y_2 - y_3}{v_3}) m_4(\frac{y_3}{v_4}) dy_3 \Big) dy_2 \Big) dy_1 \tag{8}$$

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 $y_3 = v_4x_4$, $y_2 = v_3x_3 + y_3$, and $y_1 = v_2x_2 + y_2$. This allows us to perform the integrands one at a time. The transformation of variables used here works for any vector $\vec{v}$. However, there are many possible transformations. Clever choice of 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.

If $v_i = \pm 1$ for all i, and messages are represented as uniform-width histograms, then each integrand in equation 8 can be reduced to a $\mathcal{O}(M \log M)$ computation using discrete Fourier transforms as in [3]. 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 8 can be closely approximated in linear time using the distance transform methods described in [3].

Linear potential functions of the form in equation 5 are quite flexible. In fact, for any multivariate potential function $\phi(\vec{x})$, we can approximate ϕ by a product of linear potential functions $\phi(\vec{x}) \approx \prod_i \phi_i(\vec{x} \cdot \vec{v}_i)$. To see this, note that the integral transform

$$\mathcal{R}^\dagger[\psi] = \int_{|v|=1} \psi(\vec{x} \cdot \vec{v}, \vec{v})\, d\vec{v} \tag{9}$$

is the adjoint of the Radon transform [2]. Since the Radon transform is invertible, its adjoint must also be invertible. This means that for any potential function $\phi(\vec{x})$, there exists a function $\psi(r, \vec{v})$ such that $\log \phi(\vec{X}) = \mathcal{R}^\dagger[\psi]$. Thus, given enough unit vectors $\vec{v_i}$, $\log \phi(\vec{x})$ can be represented almost exactly by a sum $\sum_i \log \phi_i(\vec{x} \cdot \vec{v_i})$.

If the nonlinearity g is a delta function, then the first integral in equation 8 is not necessary. We refer to such factor nodes as *hard linear constraint nodes*. Hard linear constraint nodes are useful because they allow overcomplete representations of the problem space $\vec{X}$ to be used in graphical models. 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.

When the representation of $\vec{X}$ is overcomplete, then there are 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 3), enforcing such constraints would be intractable. Using the methods in equation 8, these constraints can be efficiently enforced using a set of hard linear constraint nodes. Section 5 gives an application of hard linear constraint nodes.

3.2. Nonlinear Constraint Nodes

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

$$\phi(\vec{x}) = g(g_1(x_1) + \cdots + g_N(x_N)) \tag{10}$$

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 3 to get:

$$M_1(x_1) = \iint g(g_1(x_1) + g_2(x_2) + g_3(x_3)) \; m_2(x_2)\, m_3(x_3)\, dx_2\, dx_3 \tag{11}$$

$$= \iint g(g_1(x_1) + \hat{x}_2 + \hat{x}_3) \hat{m}_2(\hat{x}_2) \hat{m}_3(\hat{x}_3) d\hat{x}_2 d\hat{x}_3 \tag{12}$$

$$\text{where} \quad \hat{m}_i(\hat{x}_i) = m_i(g_i^{-1}(\hat{x}_i)) \frac{\partial}{\partial \hat{x}_i} g_i^{-1}(\hat{x}_i) \tag{13}$$

We can then apply the methods of section 3.1 to get

$$M_1(x_1) = \int g(g_1(x_1) + y_1) \int \hat{m}_2(y_1 - y_2) \hat{m}_3(y_2)\, dy_2\, dy_1 \tag{14}$$

where we have made the change of variables $y_1 = \hat{x}_2 + \hat{x}_3$ and $y_2 = \hat{x}_3$.

If g_i is not invertible, we can still apply the same technique if we integrate equation 3 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.

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 [14]. Since additional nonlinear potential functions of pairs of variables y_i and y_{i+1} can be embedded into equation 14 at no additional computational cost, many non axis-aligned Gaussians and other potential functions can also be computed efficiently using these methods.

3.3. 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 [5]. The models we will be exploring later in this paper have both of these problems. In fact, we have found empirically that standard loopy belief propagation typically fails to converge for the model of figure 3, even using a variety of dampening, scheduling, and 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 [21]. This has lead to the development of belief propagation algorithms that minimize the Bethe free energy directly [22, 6], and do so while ensuring convergence.

In the examples presented here, we use the algorithm described in [6], which modifies equations 2 and 3 by:

$$m^t_{i\to f}(x_i) = m^t_{f\to i}(x_i)^{\frac{1-n_i}{n_i}} \prod_{g\in\mathcal{N}(i)\setminus f} m^{t-1}_{g\to i}(x_i)^{\frac{1}{n_i}} \tag{15}$$

$$m^t_{f\to i}(x_i) = \sum_{\vec{x}_{\mathcal{N}(f)\setminus i}} \left(\tilde{\phi}_f\left(\vec{x}_{\mathcal{N}(f)}\right) \prod_{j\in\mathcal{N}(f)\setminus i} m^t_{j\to f}(x_j) \right) \tag{16}$$

$$\tilde{\phi}_f(\vec{x}_{\mathcal{N}(f)}) = \phi_f(\vec{x}_{\mathcal{N}(f)}) \prod_{j\in\mathcal{N}(f)} b^\tau_j(x_j)^{\frac{n_j-1}{n_j}} \tag{17}$$

where $b^\tau_j(x_j)$ is the belief at variable node j the last time the algorithm converged. Once convergence is reached, $b^\tau_j(x_j)$

is updated according to $b_i^t(x_i) \propto \prod_{g \in \mathcal{N}(i)} m_{g \to i}^t(x_i)^{\frac{1}{n_i}}$ and the algorithm continues until $b_j^\tau(x_j)$ 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 Heske's 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 [22], which introduces a temperature T, and replaces the energy function of equation 1 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.

3.4. Nonparametric Message Representation

For continuous random variables, the integrals of equation 6 or 8 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. However, 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. Using histograms with many bins can help to alleviate this error, but only at the cost of computational efficiency and working memory requirements.

Some applications have achieved good results by representing messages using a single Gaussian [13, 20]. However, for many vision applications, marginals are often highly non-Gaussian. For example, in the application of section 5, messages are typically bimodal.

Another approach, referred to as particle-based or nonparametric belief propagation (NBP) [18, 8] is to represent each message as a mixture of Gaussians. Belief propagation for real-valued variables using these techniques has been shown to achieve superior performance and with much greater efficiency than discrete representations. However, these methods were developed for pairwise connected Markov Random Fields using standard belief propagation (equations 2 - 4). Both higher-order potential functions and Heske's convergent belief propagation pose several additional obstacles for nonparametric belief propagation.

Nonparametric belief propagation becomes inefficient when the number of factor nodes connected to each variable node (call this D) becomes high, because it requires sampling from a product of D different mixtures of several Gaussians. The method is typically considered impractical for $D > 8$ [8]. Graphical models with higher order potential functions tend to have high values of D. For instance, the denoising problem in section 4 uses a network with $D = 12$. This problem is exacerbated by the corrections applied in equation 17. Finally, the exponent introduced in equation 15 means that instead of sampling from a product of D mixtures of Gaussians (already a difficult problem for large D), one must instead sample from such a product *raised to an arbitrary exponent* (a far more difficult problem).

The use of variable-width bin histograms (where each bin may have a different width) has been used successfully to boost the performance of the join tree algorithm [9]. Here we show that such a representation, when applied to belief propagation, can overcome the obstacles encountered in applying particle-based representations to Heske's guaranteed-convergent LBP variation [6], or to problems with highly-connected graphs. We require that each message $m_{i \to f}(x_i)$ at a given variable node i must have the same bin edges. Because of this, and because histogram bins are non-overlapping (unlike Gaussian kernels), both multiplication and exponentiation now become trivial:

$$\left((\sum_{k=1}^{M} \alpha_k h_k(x))(\sum_{k=1}^{M} \beta_k h_k(x)) \right)^\eta = \sum_{k=1}^{M} (\alpha_k \beta_k)^\eta h_k(x) \tag{18}$$

where $h_k(x)$ is 1 when x is within the range of bin k, and zero otherwise. Thus, equation 15 can be computed efficiently, even for high values of D.

The edges of each bin can be updated after the completion of each inner-loop, when $b_j^\tau(x_i)$ is updated. In this paper, the bin edge locations are adjusted so that the likelihood of each bin (according to beliefs $b(x_i)$) is roughly equal. Updating the locations of bin edges requires the corrections $b_j^\tau(x_i)$ be interpolated, but the variable's beliefs and messages can remain unchanged until they are replaced in the next update. Note that the use of adaptive histograms is related to approaches to improve the efficiency of LBP by adaptively restricting the search space to only those states with high predicted likelihoods [1].

4. Application: Higher-Order Spatial Priors

Several state-of-the-art computer vision algorithms use belief propagation. A number of these, including stereo [19] and photometric stereo [20] 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 complexities 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 stronger prior image models 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 [17]. FoE has shown itself to be highly effective at capturing complex image statistics by

a) Original Input | **b) Noisy Image** ($\sigma = 20$) PSNR = 21.11 | **c) Pairwise MRF** PSNR = 27.03 | **d) High-Order** 2×2 **Factors** PSNR = 28.83

Figure 1. Using higher-order Fields of Experts to perform image denoising. **a)** A cropping from the original image (from [14]). **b)** The original image with additive Gaussian noise of $\sigma = 20$. **c)** The output of a pairwise-connected Markov Random Field, similar to the model described in [3]. Pairwise models tend to produce piecewise constant image regions [10]. **d)** Results using three 2×2 Field of Experts filters learned from natural images, using the graphical model of figure 2 and the methods described in section 3.

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:

$$p(\vec{I}) \propto \prod_{C} \prod_{i=1}^{K} \left(1 + \frac{1}{2}(\vec{I}_C \cdot \vec{J}_i)^2\right)^{-\alpha_i} \qquad (19)$$

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 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 [10]. The authors showed that using three 2×2 Field of Experts filters yields a significant improvement over pairwise models. In their approach, the authors mitigate the computational complexity of equation 3 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 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.

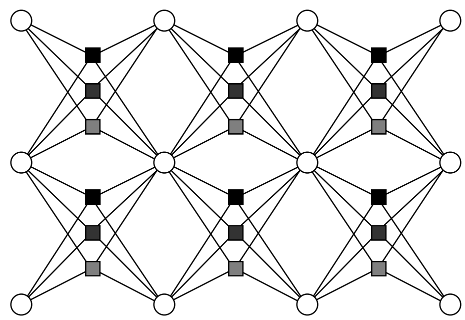

Figure 2. A factor graph used to perform image denoising using three 2×2 Fields of Experts filters. Each variable node, shown here as circles, represents the true image intensity at a given pixel. The three gray squares represent factor nodes corresponding to the three 2×2 Fields of Experts filters.

Using the methods of section 3, 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 [10]. 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 that has been corrupted with additive Gaussian noise of known variance. The object is to remove this noise and recover the original image. The Field of Experts spatial prior is implemented according to the factor graph in figure 2. Here, the Gaussian likelihood is absorbed into the factor nodes neighboring each pixel, and requires no additional messages. Note that the model is capable of performing denoising in a variety of other noise circumstances, such as non-Gaussian or multiplicative noise.

Noise Level	Pairwise MRF from [10]	2×2 Model from [10]	2×2 Model from Fig. 2
$\sigma = 10$	30.73	30.89	31.62
$\sigma = 20$	26.66	27.29	27.40

Table 1. Peak signal-to-noise ratio for pairwise and higher-order models, averaged over ten images.

Results from our approach are shown in figure 1. As shown in figure 1**c**, pairwise-connected Markov random fields tend to produce piecewise constant results [10]. Results using higher order factor nodes (figure 1**d**) are able to overcome this limitation. We measured the mean peak signal to noise ratio (PSNR) for these results over the same set of 10 images from the Berkeley segmentation database [12] that was used in [10]. Here, $PSNR = 20\log_{10}(255/\sqrt{MSE})$, where MSE is the mean squared error. These results are shown in table 1. Our results show a significant improvement over pairwise connected MRFs. We also record an improvement over the higher-order Fields of Experts model described in [10], which uses the same statistical model of natural images. This is most likely due to the convergent variant of belief propagation [5] and non-parametric message representations used here.

More importantly, using the methods of section 3, 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 on a 256×256 image. By comparison, the method of [10] took 16 minutes per iteration on a 3GHz Xeon, and benefited from a reduced search space. Convergence typically required around 30 iterations.

5. Application: 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 [7]. 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). 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:

$$i(x, y) = \max(0, \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}}) \quad (20)$$

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

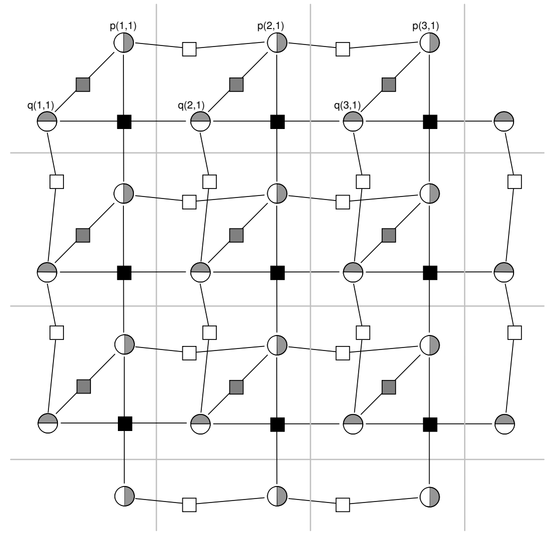

Figure 3. Shape-from-shading factor graph for a 3×3 image. Variable nodes are shown as circles, and factor nodes as shown as squares. Variable nodes include nodes for $p = \frac{\partial z}{\partial x}$ and $q = \frac{\partial z}{\partial y}$. Factor nodes include Lambertian constraint nodes (gray),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

In figure 3, 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. Specifically, an identity holds that

$$p(x, y) - q(x, y) + q(x+1, y) - p(x, y+1) = 0 \quad (21)$$

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. Satisfying these constraints has historically been problematic for SFS. Using the methods of section 3, we can enforce these linear dependencies efficiently using hard linear constraint nodes. These integrability constraint nodes are shown in figure 3 as black squares. These integration nodes are similar to those used in [13], 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 3 are required to perform belief propagation at these nodes efficiently.

The square nodes shown in gray in figure 3 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 $P(p, q|i) = const$ whenever equation 20 holds, and zero otherwise. However, any reflectance function can be used. $\phi_L(p, q)$ could easily be changed to han-

a) Original Image | **b) Linear Constraint Nodes** Mean Squared Error = 108 | **c) Lee & Kuo [11]** Mean Squared Error = 3390 | **d) Zheng & Chellappa [24]** Mean Squared Error = 4240

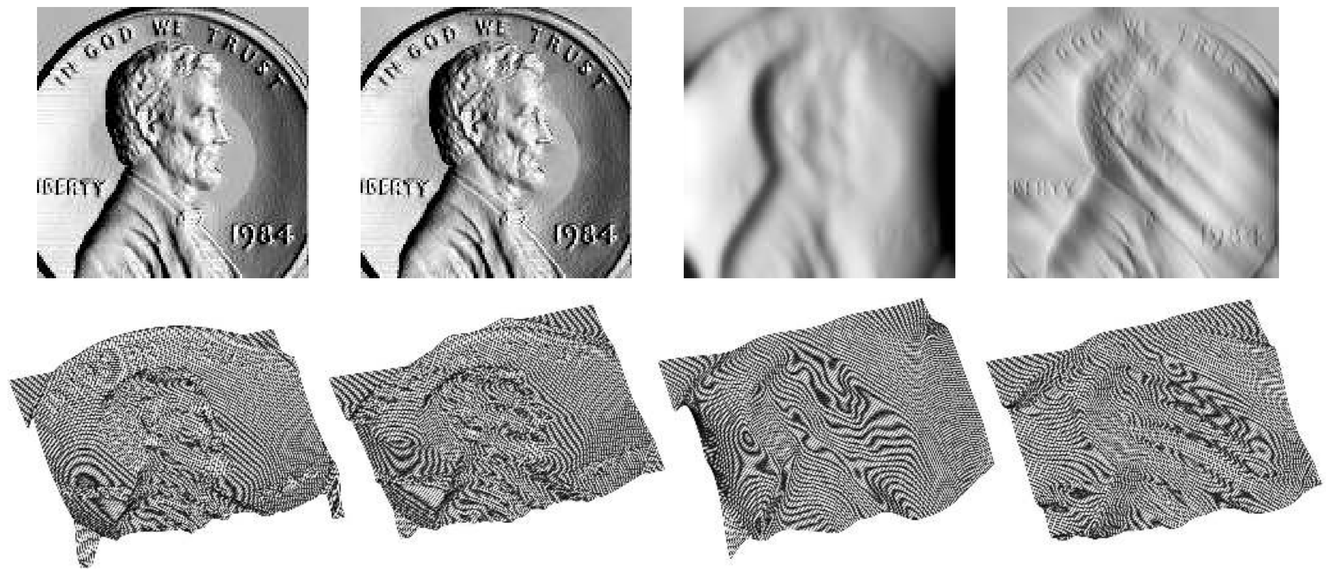

Figure 4. Comparing our SFS results (column b) with previous energy-minimization approaches (columns c & d). Each column contains a 3D wire mesh plot of the surface (bottom) and a rendering (top) of that surface from a light source at location $(1, 0, 1)$, using the Lambertian reflectance equation. **a)** The original 128×128 surface [23]. 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 [11] (source code obtained from [23]). This algorithm performed best out of six SFS algorithms reviewed in the survey paper [23]. **d)** The surface recovered using the method described by Zheng and Chellappa [24] (source code obtained from [23]). It is important to note that re-rendering the surface output from our algorithm (column b) 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).

dle specular surfaces, or even scenes with multiple or diffuse light sources. Furthermore, specifying nondeterministic $\phi_L(p, q)$ 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. For example, 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.

Many shape from shading methods handle this ambiguity by assuming that the surface shape is known along the image border [23], or by computing the maximal surface, where each point is as close to the observer as possible [16]. If these additional assumptions are met, such techniques can recover the 3D shape fairly well. However, a more flexible approach 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 could 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. [11], [24]). 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 [23]. Belief propagation methods have proven themselves more robust to local minima, making 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:

$$p(Z) \propto \prod_{p,q} exp(-\frac{|p| + |q|}{\sigma_1}) exp(-\frac{|\Delta p| + |\Delta q|}{\sigma_2}) \quad (22)$$

Priors of the form $exp(-\frac{|p|}{\sigma_1})$ can be absorbed into the factor nodes adjacent to each variable, and so they require no additional message passing. Priors of the form $exp(-\frac{|\Delta p|}{\sigma_1})$ are implemented using an additional set of factor nodes. These nodes are shown as white squares in figure 3.

In figure 4, we show the results of our SFS model. We also compare our results with previous energy-minization methods [11, 24]. Our approach offers a noticeable 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 22. Further improvement to the results of this approach can only be achieved by improving the model of the 3D surface priors (equation 22).

Note that the model of the 3D surface priors used in this section use the same weak pairwise form that we improve upon in section 4. 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 section 4. This can be expected to improve our results still further. Such a spatial prior might also be highly useful for stereo [19], photometric stereo [20], and other forms of depth inference.

6. Conclusions

In this paper, we present methods that are capable of making belief propagation both efficient and successful for a powerful class of graphical models that were previously intractable. To demonstrate this technique, we use our methods to perform inference over rich spatial priors, and to solve the problem of shape-from-shading. In addition to improved performance, SFS approached using LBP shows the potential to generalize to more complex problems, such as handling uncertainty in surface albedo and lighting conditions, exploiting a strong spatial prior (using the methods of section 4), and utilizing additional depth information when available, such as stereo [19], occlusion contours, or sparse, low-resolution range scans [15]. In general, the ability to accurately perform inference in Markov random fields with higher-order, non-pairwise cliques and real-valued variables may prove to be highly useful for solving a variety of computer vision problems.

Acknowledgements: I thank Tai Sing Lee for his support and advice, and S. Roth & the authors of [10] for graciously sharing filters & results. This research was funded by an NSF Graduate Research Fellowship to the author and NSF IIS-0413211 to TS Lee.

References

[1] J. Coughlan and H. Shen. Shape matching with belief propagation: Using dynamic quantization to accomodate occlusion and clutter. In *CVPRW*, page 180, 2004. 4

[2] S. R. Deans. *The Radon Transform and Some of Its Applications*. John Wiley & Sons, 1983. 3

[3] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, 2004. 2, 5

[4] W. T. Freeman, E. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int. J. Comp. Vis.*, 40(1):25–47, 2000. 1, 2

[5] T. Heskes. On the uniqueness of loopy belief propagation fixed points. *Neural Comp.*, 16(11):2379–2413, 2004. 3, 6

[6] T. Heskes, K. Albers, and B. Kappen. Approximate inference and constrained optimization. In *UAI*, 2003. 1, 2, 3, 4

[7] B. K. P. Horn. Obtaining shape from shading information. pages 123–171, 1989. 6

[8] M. Isard. Pampas: Real-valued graphical models for comnputer vision. In *CVPR*, pages 613–620, 2003. 4

[9] A. Kozlov and D. Koller. Nonuniform dynamic discretization in hybrid networks. In *UAI*, pages 314–32, 1997. 4

[10] X. Lan, S. Roth, D. P. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV*, pages 269–282, 2006. 1, 5, 6, 8

[11] K. Lee and C. Kuo. Shape from shading with a linear triangular element surface model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(8):815–822, 1993. 7, 8

[12] 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. 6

[13] N. Petrovic, I. Cohen, B. J. Frey, R. Koetter, and T. S. Huang. Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In *CVPR*, pages 743–748, 2001. 1, 4, 6

[14] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003. 3, 5

[15] B. Potetz and T. S. Lee. Scaling laws in natural scenes and the inference of 3D shape. In *NIPS*, pages 1089–96. 2006. 8

[16] E. Prados, F. Camilli, and O. Faugeras. A unifying and rigorous shape from shading method adapted to realistic data and applications. *J. Mathematical Imaging and Vision*, 25(3):307–328, 2006. 7

[17] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005. 4

[18] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *CVPR*, 2003. 4

[19] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, 2003. 1, 2, 4, 8

[20] K. L. Tang, C. K. Tang, and T. T. Wong. Dense photometric stereo using tensorial belief propagation. In *CVPR*, pages 132–139, 2005. 1, 2, 4, 8

[21] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000. 2, 3

[22] A. L. Yuille. An algorithm to minimize the Bethe free energy. In *EMMCVPR*, 2001. 3, 4

[23] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, 1999. 7

[24] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(7):680–702, 1991. 7, 8