Who Killed the Directed Model?

Justin Domke, Alap Karapurkar, and Yiannis Aloimonos
Department of Computer Science
University of Maryland
{domke,karapurk,yiannis}@cs.umd.edu

Abstract

Prior distributions are useful for robust low-level vision, and undirected models (e.g. Markov Random Fields) have become a central tool for this purpose. Though sometimes these priors can be specified by hand, this becomes difficult in large models, which has motivated learning these models from data. However, maximum likelihood learning of undirected models is extremely difficult- essentially all known methods require approximations and/or high computational cost.

Conversely, directed models are essentially trivial to learn from data, but have not received much attention for low-level vision. We compare the two formalisms of directed and undirected models, and conclude that there is no a priori *reason to believe one better represents low-level vision quantities. We formulate two simple directed priors, for natural images and stereo disparity, to empirically test if the undirected formalism is superior. We find in both cases that a simple directed model can achieve results similar to the best learnt undirected models with significant speedups in training time, suggesting that directed models are an attractive choice for tractable learning.*

1. Introduction

Low-level perception involves dealing with large amounts of uncertainty. Many problems can be phrased as "inference", or using prior knowledge to predict unseen quantities. Though it is possible to specify simple priors by hand, there is great interest in the use of data to learn priors that are too complex to construct manually.

For low-level vision, the most common type of priors have been formulated as undirected models (e.g. Markov random fields, or conditional random fields). These have been successfully learnt as priors for natural images [13, 22, 23], stereo depth [16, 15], single image depth [14], object segmentation [2], optical flow [12], intrinsic images [20], etc. However, undirected models are in general extremely difficult to learn. Essentially all uses of undirected models have either restricted the model to a simple (e.g. Gaussian or Laplacian) form, used variational approximations, or used exhaustive Markov chain Monte Carlo sampling. (Training time of days or weeks is not uncommon for MCMC based learning.) In short, exact maximum likelihood learning of undirected models seems to require repeated inference, which is often unacceptably slow. Approximations, on the other hand, can be difficult to generalize to new situations. These difficulties have limited the complexity of models that are practical to learn.

Directed models, an alternative type of graphical model, do not suffer from many of the difficulties of undirected models, but have received comparatively little attention for low-level vision [8, 7]. This may be due, in part, to the fact that directed models require imposing an order on variables, that is unnatural for quantities like images. Nevertheless, directed models have major computational advantages– there is great flexibility in specifying the prior while permitting exact maximum likelihood learning.

As we will argue, there is no *theoretical* reason to believe that any particular low-level vision prior is better represented by either type of model. The traditional view that undirected models better represent low-level vision may very well be correct. However, before dismissing the easier directed models, to investigate an *experimental* question. How good are the priors we can estimate using directed models, when compared to the state of the art in undirected model learning?

To this end, we develop simple priors for two quantities: stereo disparity, and natural images. In both cases, exact maximum likelihood learning is easy and proceeds in a reasonable amount of time. For natural images, we fit the conditional probability of one pixel given some neighbors by a mixture of Gaussians. For stereo, we learn a discrete representation conditioning each pixel's disparity on neighboring disparities, as well as image information near that pixel. Despite the simplicity of these approaches, in both cases we find that the priors are able to produce results comparable to state of the art learnt undirected models.

The basic thesis of this paper is that directed models can

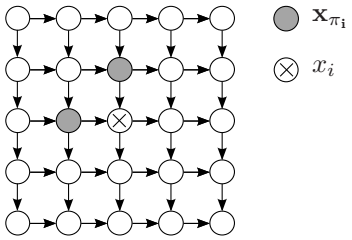

Figure 1. An example graph where the parents of a variable consist of its left and upper neighbors.

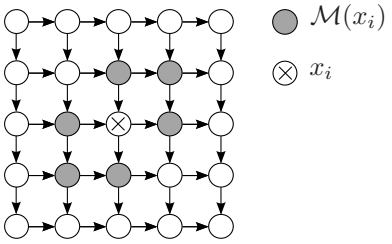

Figure 2. The Markov blanket $\mathcal{M}$ for x_i. This consists of x_i's parents, and children, as well as its childrens' parents.

also represent priors for low-level vision, while enjoying major computational advantages for learning.

2. Background

Section 2 reviews well known material on graphical models. The expert reader can skip to section 3.

2.1. Directed Models

By elementary rules of probability, any probability distribution can be exactly written as a product of terms, where each term is the conditional probability of one variable, given all those before it in some order.

$$p(x_1, x_2, ..., x_N) = p(x_1)p(x_2|x_1)...p(x_N|x_1, x_2, ...x_{N-1}) \quad (1)$$

However, unless the number of variables is small, this representation will usually be impractical, since most terms involve conditional probabilities defined over large numbers of variables. In a directed model, one assumes a set of "parents" for each variable, that render that variable independent all all others before it in the ordering. Let π_i denote the set of parents for variable i. Then the assumption is that

$$p(x_i|x_1, ..., x_{i-1}) = p(x_i|\mathbf{x}_{\pi_i}) \quad (2)$$

and so

$$p(x_1, x_2, ..., x_N) = \prod_i p(x_i|\mathbf{x}_{\pi_i}). \quad (3)$$

Given a set of assumed parents for each node, it is often convenient to picture the situation by drawing a graph with one node for each variable, and directed edges from each parent to each child (Fig. 1).

Notice that there is no reference in Eq. 2 to any conditional independence to nodes $x_{i+1}, x_{i+2}, ..., x_N$. A directed model asserts only that x_i is independent of $\{x_1, x_2, ..., x_{i-1}\}$ given x_{π_i}. So what is the "Markov blanket" of x_i, i.e., the set of variables that render it conditionally independent of *all* others? This turns out to consist of x_i's parents, children, and its childrens' parents (Fig. 2).

Now, we turn to the issue of learning. Suppose we have a set of samples $\{\hat{\mathbf{x}}\}$ from some true (but unknown) distribution $p(\mathbf{x})$. The most common way to do learning is by maximum likelihood. For some candidate distribution $q(\mathbf{x})$ define the log-likelihood to be

$$l(q) = \sum_{\hat{\mathbf{x}}} \log q(\hat{\mathbf{x}}). \quad (4)$$

Different justifications are sometimes given for the maximum likelihood criterion. For our purposes here, however, we consider maximizing the likelihood to be a surrogate for minimizing the KL-divergence to the true distribution, $KL(p||q)$. An informal motivation for this is that

$$\arg\min_q KL(p||q) = \arg\min_q \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \quad (5)$$

$$= \arg\max_q \int p(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x} \quad (6)$$

$$\approx \arg\max_q \sum_{\hat{\mathbf{x}}} \log q(\hat{\mathbf{x}}). \quad (7)$$

So in the high data limit, the maximum likelihood solution will converge to the minimum KL-divergence solution (under certain conditions [21]). The reason that directed models are comparatively easy to learn is that the likelihood "decomposes" into a sum of terms. Substituting Eq. 2 into Eq. 4 yields

$$l(q) = \sum_i \sum_{\hat{\mathbf{x}}} \log q(\hat{x}_i|\hat{\mathbf{x}}_{\pi_i}). \quad (8)$$

So given q, the likelihood can be computed in closed form. In general, to maximize this expression with respect to q does not present any particular difficulties.

For low-level vision it is common to impose translation invariance, to reduce model complexity, and the required amount of training data. This means that only a single function $q(x_i|\mathbf{x}_{\pi_i})$ needs to be learned.

2.2. Undirected Models

In an undirected model, or Markov random field, one directly specifies the Markov blanket of each variable. Denote by $\mathcal{N}(i)$ the set of neighbors of i. One then asserts that x_i is independent of all other variables given its set of neighbors.

$$p(x_i|x_1, ..., x_{i-1}, x_{i+1}, ..., x_N) = p(x_i|\mathbf{x}_{\mathcal{N}(i)}) \quad (9)$$

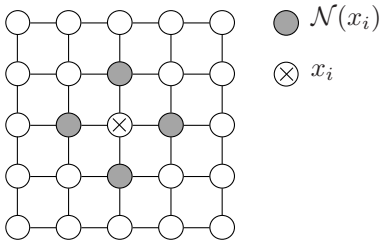

Figure 3. The classic undirected model where a variable is independent of all others given its four neighbors.

The neighborhood system must be symmetric– $j \in \mathcal{N}(i)$ if and only if $i \in \mathcal{N}(j)$. An undirected model is pictured by drawing a graph with one node for each variable, and undirected edges between the nodes for all variables that are neighbors. (Fig. 3)

The immediate question is, given an undirected model, what form can its probability distribution take? It is not easy to specify $p(\mathbf{x})$ in terms of local conditional probability distributions, because in general graphs these local conditional distributions turn out to have severe, non-obvious constraints [4]. The solution ultimately came in the form of the famous Hammersley Clifford theorem[4], which applies only to positive probability distributions.

Theorem: $p(\mathbf{x}) > 0$ obeys the set of conditional independencies asserted by a graph if and only if there exist functions $f_C(\mathbf{x}_C)$ such that

$$p(\mathbf{x}) = \frac{1}{Z} \exp(\sum_C f_C(\mathbf{x}_C)), \tag{10}$$

where the sum is over the set of *cliques* C in the graph, and $Z = \int \exp(\sum_C f_C(\mathbf{x}_C)) d\mathbf{x}$.

So, specifying a valid distribution $p(\mathbf{x})$ is equivalent to specifying the set of functions $f_C(\mathbf{x}_C)$.

Now, suppose we would like to do maximum likelihood density estimation of an undirected model. Substituting Eq. 10 into 4 gives

$$l(f) = \sum_{\mathbf{x}} (\sum_C f_C(\mathbf{x}_C) - \log Z). \tag{11}$$

Unfortunately, it is in general not possible to compute $l(f)$. The difficulty is the presence of the normalization constant. Computing Z requires a high dimensional integral (or sum for discrete variables). Except in some special cases, there is no tractable method to exactly find Z. In fact, for discrete variables, finding Z is known to be NP-hard. Since it is difficult to even compute the likelihood, it is unsurprising that maximizing it is also very difficult. There are algorithms for learning discrete models, such as Iterated Proportional Fitting. However, these algorithms require repeatedly computing marginal distributions, which is also known to be NP-hard (to do exactly) in general graphs.

Given the above difficulties, work on learning undirected models either restricts attention to simple (e.g. Gaussian) functions [18], uses exhaustive Markov chain Monte Carlo techniques [13, 23], variational methods [19, 22], the pseudo-likelihood approximation [5], and/or other approximations [17] such as contrastive divergence[13]. While these methods have proven successful in their domains, it is often difficult to generalize these results, due to either computational requirements, or domain-specific approximations.

2.3. Directed vs. Undirected Models

Computational considerations in the learning stage clearly favor directed models. Exact maximum likelihood learning in general undirected models is intractable, while presenting no particular difficulty for directed models. In the case of discrete variables a well-known result shows maximum likelihood learning for a directed model often reduces to setting each conditional probability distribution equal to the observed frequencies.

Absent experiment, it is not clear which type of model can better represent a given type of prior. Each model makes assertions of conditional independence that are unlikely to ever be exactly satisfied. However, as argued above, maximum likelihood learning of a directed or undirected model (if feasible) will converge to the closest representable distribution, in the sense of KL-divergence.

One common misconception about directed models stems from interpreting the model causally. It does not seem to make sense to think of the pixels in the upper left-hand corner of an image "causing" the pixel intensities in the center of an image. However, using a directed model makes no such assumptions. To take an extreme example, imagine that each pixel is conditioned on all those before it, as in Eq. 1. By definition, this can exactly represent any distribution. Limiting the number of parents to each node reduces the space of representable distributions, but so does limiting the number of links in an undirected model. See Pearl [10] for more details on causality.

In summary, both undirected and directed models impose assumptions of conditional independence that are unlikely to be exactly true in practice. In both cases, maximum likelihood learning results in the closest distribution, in the sense of KL-divergence. However, in general, maximum likelihood learning of undirected models presents severe computational difficulties. It is possible that undirected models truly do allow a more accurate approximation of some priors. However, this can only be determined by experiment. Even if this was the case, the computational advantages of directed models would still need to be weighed against these representational issues.

3. Image Prior

Our image model is pictured in Fig. 4. Essentially, each pixel is conditioned on the twelve pixels before it in the five

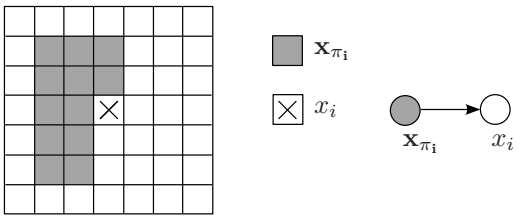

Figure 4. The model for natural images, $p(x_i|\mathbf{x}_{\pi_i})$. Here, x_i denotes the intensity of pixel i.

by five surrounding patch.

We assume the prior should be translation invariant. Hence, we fit a single function $p(x_i|\mathbf{x}_{\pi_i})$ that approximates the conditional probability of any one pixel x_i given its neighboring pixels $\mathbf{x}_{\pi_i}$; this single function applies to any position in the image. Thus, for simplicity, the index i is dropped in the following discussion.

3.1. Representation

The image prior is represented as a conditional mixture of Gaussians. Suppose that the *joint* density over x and $\mathbf{x}_\pi$ is represented by some mixture of Gaussians.

$$p(x, \mathbf{x}_\pi) = \sum_m \alpha_m \mathcal{N}(\begin{bmatrix} x \\ \mathbf{x}_\pi \end{bmatrix}, \Sigma_m, \mu_m) \tag{12}$$

It is well known that the marginal of this over $\mathbf{x}_\pi$ will be

$$p(\mathbf{x}_\pi) = \sum_m \alpha_m \mathcal{N}(\mathbf{x}_\pi, \Sigma_m^*, \mu_m^*) \tag{13}$$

where μ_m^* denotes the vector containing all elements of μ_m except the first, and Σ_m^* denotes the submatrix of Σ_m containing all elements except the first column and first row [11].

The conditional distribution follows immediately from Eqs. 12 and 13.

$$p(x|\mathbf{x}_\pi) = \frac{\sum_m \alpha_m \mathcal{N}(\begin{bmatrix} x \\ \mathbf{x}_\pi \end{bmatrix}, \Sigma_m, \mu_m)}{\sum_m \alpha_m \mathcal{N}(\mathbf{x}_\pi, \Sigma_m^*, \mu_m^*)} \tag{14}$$

3.2. Learning

To do maximum likelihood density estimation, we should now fit the distribution to maximize the conditional likelihood

$$\sum_{\{(\hat{x}, \hat{\mathbf{x}}_\pi)\}} \log p(\hat{x}|\hat{\mathbf{x}}_\pi) = \sum_{\{(\hat{x}, \hat{\mathbf{x}}_\pi)\}} (\log p(x, \mathbf{x}_\pi) - \log p(\mathbf{x}_\pi)), \tag{15}$$

where $p(x, \mathbf{x}_\pi)$ and $p(\mathbf{x}_\pi)$ are given in Eqs. 12 and 13. The derivatives of Eq. 15 with respect to α_m, μ_m, and Σ_m are easily derived in terms of the derivatives of the logarithm of a mixture of Gaussians[11].

σ	$\lambda(\sigma)$	Lena	Barbara	Boats	House	Peppers	Mean	R&B
1	1.5	48.2	48.2	48.1	48.9	48.2	**48.3**	47.90
2	1.5	42.8	42.8	42.2	43.9	42.9	42.9	**43.02**
5	2	37.7	36.7	36.1	37.9	37.5	37.2	**37.49**
10	2.5	34.4	32.2	32.8	34.8	33.8	33.6	**34.05**
15	3	32.6	29.7	31.0	33.3	31.7	31.7	**32.04**
20	3.5	31.4	27.9	29.8	32.1	30.3	30.3	**30.57**
25	4.5	30.4	26.5	28.8	31.1	29.2	29.2	**29.38**
50	9	27.5	23.2	25.7	27.9	25.6	**26.0**	25.09
75	12.5	25.7	22.3	24.1	25.9	23.4	**24.3**	22.76
100	15.5	24.5	21.7	22.9	24.4	22.0	**23.1**	20.74

Table 1. Denoising Results, including the extra parameters $\lambda(\sigma)$. R&B refers to the Roth and Black Field of experts model [12], which uses similar parameters (see text).

Before learning, we initialized α_m uniformly, and initialized all Σ_m and μ_m to be the covariance and mean of the training data, with some slight perturbation to break symmetries. We used 1,000,000 randomly sampled 5×5 patches, taken from the same Berkeley segmentation database used to learn some other recent priors[13, 22]. We learned three priors, with mixtures of one, two, and five components.

The mixtures were learnt using stochastic gradient ascent, with a small fixed step size. In order to be a valid mixture of Gaussians, the matrices Σ_m must be constrained to be positive definite, and the parameters α_m must be constrained to be positive and sum to one. This is handled by reparameterizing $\Sigma_m = M_m M_m^T$ and $\alpha_m = \exp(\theta_m)/\sum_n \exp(\theta_n)$. We made 100 passes through the data, with the order of the data randomly shuffled before each pass. For computational purposes in our interpreted implementation, the samples were considered in batches of size 1000. In our Matlab implementation on a 2.79 GHz PC, learning took about one hour for the five component prior.

3.3. Denoising Experiments

As is common in image denoising, we assume that an image has been corrupted with Gaussian noise of known variance. Then, given the observed noisy image $\mathbf{w}$, the posterior distribution over the noiseless image $\mathbf{x}$ is

$$p(\mathbf{x}|\mathbf{w}) \propto p(\mathbf{w}|\mathbf{x}) p(\mathbf{x}) \tag{16}$$

where $p(\mathbf{w}|\mathbf{x}) \propto \exp(||\mathbf{w} - \mathbf{x}||^2/(2\sigma^2))$, and $p(\mathbf{x})$ is the prior over natural images learned above. Then it is possible to find a local maximum of $\log p(\mathbf{w}|\mathbf{x}) + \log p(\mathbf{x})$ using a nonlinear optimization to produce a denoised image $\mathbf{x}$.

The gradient of $\log p(\mathbf{w}|\mathbf{x})$ is well known. The gradient of $\log p(\mathbf{x})$ again follows from Eq. 14 using standard formulas [11].

To maximize $p(\mathbf{x}|\mathbf{w})$ we used a quasi-Netwon method (Limited memory BFGS [1]). To reduce memory require-

σ	Lena	Barbara	Boats	House	Peppers	Mean	W&F
1	47.7	47.8	47.8	48.7	47.8	**47.9**	43.6
2	42.2	42.5	41.6	43.5	42.4	**42.4**	40.2
5	37.1	36.2	35.4	37.0	36.9	**36.5**	36.3
10	33.7	31.3	31.8	34.4	33.0	32.8	**33.3**
15	31.8	28.3	29.9	32.6	30.6	30.7	**31.1**
20	30.5	26.0	28.6	31.2	28.8	29.0	**29.4**
25	29.5	24.4	27.5	30.2	27.4	27.8	**27.9**
50	26.6	22.7	24.6	26.8	23.9	**24.9**	23.1
75	25.2	22.0	23.3	25.0	22.1	**23.5**	20.0
100	24.1	21.5	22.5	23.9	21.1	**22.6**	17.8

Table 2. Denoising Results, *not* including the extra parameters $\lambda(\sigma)$. W&F refers to the model by Weiss and Freeman [22], which also does not use such parameters.

ments, the image is divided into several patches of approximately 40 by 40 pixels each. The optimization then iteratively optimizes over the pixels in each patch. The image is first denoised with the 1-mixture prior, then the 2-mixture prior, then the full 5-mixture prior. We found that this significantly improved convergence. The image is swept over 5 times for the 1 and 2-mixture priors, and five times for the 5-mixture prior. In each sweep, 15 L-BFGS iterations are used on each patch.

Tables 1 and 2 compare the denoising results with two other image priors using a standard denoising test set. These tables give results in terms of the peak signal to noise ratio (PSNR). In general, a higher PSNR indicates a better reconstruction.

It is important to note that Roth and Black use an additional set of parameters to denoise. Rather than directly maximizing Eq. 16 they maximize $p(\mathbf{w}|\mathbf{x})^{\lambda(\sigma)}p(\mathbf{x})$ where $\lambda(\sigma)$ is some constant greater than one chosen for each noise level. Experimentally, including these constants also improves the performance of our model. We learned a similar set of parameters by testing a range of constants λ, in intervals of .5, for each noise level using a small subset of the training data images. The constant which yielded the best average PSNR scores was chosen. For completeness, we give results both with the constants (comparing to Roth and Black) and without the constants (comparing to Weiss and Freeman).

3.4. Inpainting Experiments

Figure 6 shows the results of using our image prior for image inpainting, using the data from Bertalmío et al. [3]. In each case, an image is provided, along with a mask. The pixels in the mask are optimized to maximize $p(\mathbf{x})$. Again, we first optimize using the 1-mixture prior, then the 2-mixture prior, then the 5-mixture prior. In this case, because the masks contain relatively few pixels, the optimization took place over the full image, with L-BFGS run to completion.

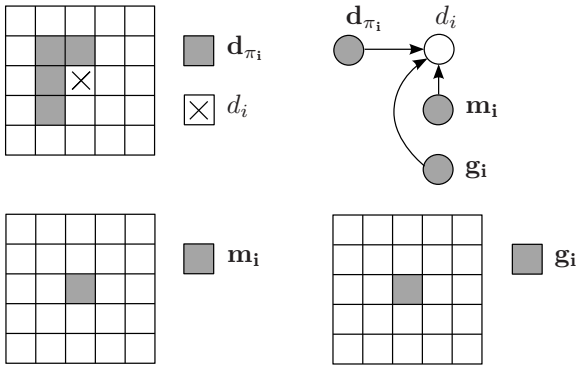

Figure 7. The model for stereo. Here, d_i denotes the disparity of pixel i, $\mathbf{m}_i$ denotes the vector of matching costs at pixel i, and $\mathbf{g}_i$ represents the gradients at pixel i.

4. Stereo

The graphical model for stereo disparity is shown in Fig. 7. We condition the disparity at each pixel d_i on the disparity of its four grayed neighbors $\mathbf{d}_{\pi_i}$, the vector of matching costs $\mathbf{m}_i$ at that pixel (explained below), and the image gradient g_i at that pixel. Since we work in a discrete disparity space, it is also convenient to discretize the variables for image gradients and matching costs. Again, we assume translational invariance, meaning only a single conditional distribution needs to be fit.

4.1. Representation

We fit $p(d_i|\mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i)$ using the parametric model

$$p(d_i|\mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i) = \frac{\exp\big(f_d(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \boldsymbol{\theta}^d) + f_m(d_i, \mathbf{m}_i, \boldsymbol{\theta}^m)\big)}{Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i)}, \tag{17}$$

where Z is the normalizing constant for the parameters $\boldsymbol{\theta}^d, \boldsymbol{\theta}^m$, and the parent values $\mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i$. Notice that Z is a function of the parent variables, but not of d.

$$Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i) = \sum_d \exp\big(f_d(d, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \boldsymbol{\theta}^d) + f_m(d, \mathbf{m}_i, \boldsymbol{\theta}^m)\big) \tag{18}$$

The functions f_d and f_m are lookup tables defined as

$$\begin{aligned} f_d(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \boldsymbol{\theta}^d) &= \theta^d_a, \quad a = \text{dcase}(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i) \\ f_m(d, \mathbf{m}_i, \boldsymbol{\theta}^m) &= \theta^m_b, \quad b = \text{mcase}(d_i, \mathbf{m}_i) \end{aligned}$$

where the functions dcase and mcase are integer valued functions that compute the "cases" for their inputs:

- **mcase** computes the cases based on matching costs. Following previous work on stereo [9], we use both the Birchfield-Tomasi matching cost[6] and gradient differences between the left and right images. At each

Figure 5. Denoising experiments with noise level $\sigma = 25$, using the adjustment $\lambda = 4.5$.

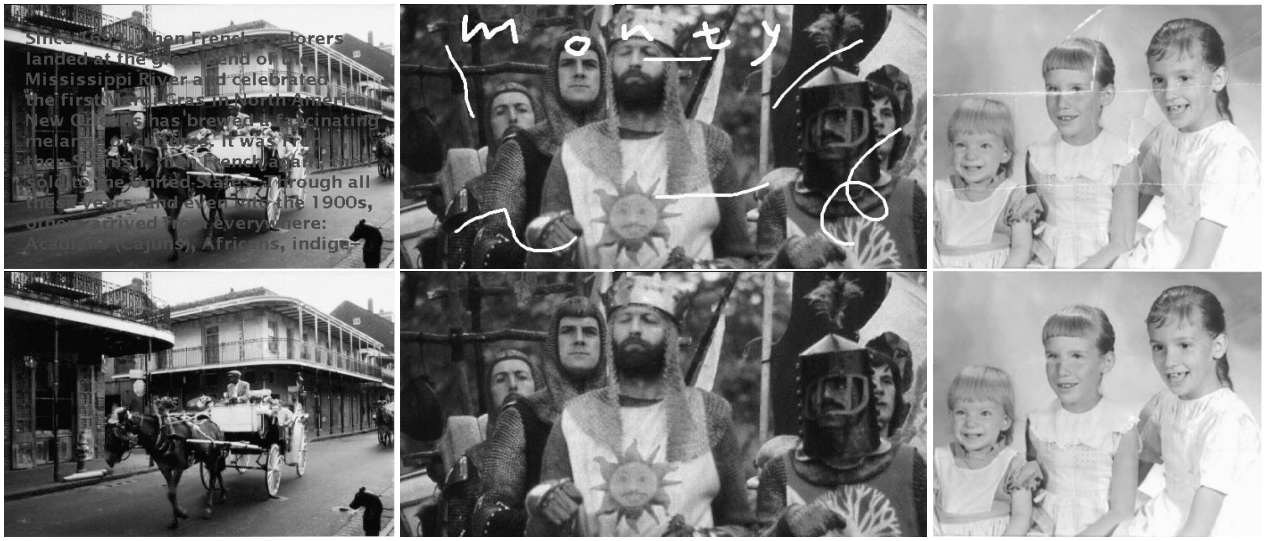

Figure 6. Inpainting experiments.

pixel i, we compute these costs for each allowed disparity, and stack these together to form the vector of matching costs $\mathbf{m}_i$. For each allowed disparity, we compute two values – the Birchfield-Tomasi matching cost, and the L_1 norm of the difference between the gradients in each image. Both costs are averaged over a 3×3 window. The two-dimensional space of intensity and gradient matching costs is discretized by binning each dimension separately. The bins are chosen such that equally many matching costs from the training data fall into each bin. We used 18 bins for the matching cost and 20 bins for the gradient cost, for a total of 360 cases. For a particular pixel, mcase$(d_i, \mathbf{m}_i)$ returns the case at disparity d_i.

- **dcase** computes the cases based on neighboring disparities and gradients. A naïve representation of all the possible disparity configurations $(d_i, \mathbf{d}_{\pi_i})$ is not practical – with 80 allowable disparities, there would be 80^5 possible cases. To reduce this, several of these cases are considered equivalent. For each parent in $\mathbf{d}_{\pi_i}$, we consider 3 cases – either d_i is equal to that parent, differs by one, or differs by greater than one. Since we are considering four parents (Fig. 7), the set of all possible disparities $d_i, \mathbf{d}_{\pi_i}$ gets divided into a more practical $3^4 = 81$ cases. The gradients $\mathbf{g}_i$ are computed by a difference filter in the horizontal and vertical directions. We used only three bins for each gradient value - $[0 - 25)$, $[25 - 50)$, and $[50 - 255]$. Since we have two gradient values (horizontal and vertical), we have $3^2 = 9$ bins for the gradient at a pixel. Finally, since there are 81 bins for $d_i, \mathbf{d}_{\pi_i}$, and 9 bins for $\mathbf{g}_i$, there are a total of 729 cases.

Theoretically, we could have used a single function with joint cases for matching costs, gradients, and disparities, but this would have required too much training data.

$$\frac{\partial}{\partial \theta_a^d} l = \sum_{\{\hat{\mathbf{d}},\hat{\mathbf{g}},\hat{\mathbf{m}}\}} \sum_i \Big(\frac{\partial}{\partial \theta_a^d} f_d(\hat{d}_i, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \boldsymbol{\theta}^d) - \frac{1}{Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \hat{\mathbf{m}}_i)} \frac{\partial}{\partial \theta_a^d} Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \hat{\mathbf{m}}_i) \Big)$$

$$\frac{\partial}{\partial \theta_b^m} l = \sum_{\{\hat{\mathbf{d}},\hat{\mathbf{g}},\hat{\mathbf{m}}\}} \sum_i \Big(\frac{\partial}{\partial \theta_b^m} f_m(\hat{d}_i, \hat{\mathbf{m}}_i, \boldsymbol{\theta}^m) - \frac{1}{Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \hat{\mathbf{m}}_i)} \frac{\partial}{\partial \theta_b^m} Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \hat{\mathbf{m}}_i) \Big)$$

$$\frac{\partial}{\partial \theta_a^d} Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \hat{\mathbf{m}}_i) = \sum_d \exp\big(f_d(d, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \boldsymbol{\theta}^d) + f_m(d, \hat{\mathbf{m}}_i, \boldsymbol{\theta}^m) \big) \frac{\partial}{\partial \theta_a^d} f_d(d, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \boldsymbol{\theta}^d)$$

$$\frac{\partial}{\partial \theta_b^m} Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \hat{\mathbf{m}}_i) = \sum_d \exp\big(f_d(d, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \boldsymbol{\theta}^d) + f_m(d, \hat{\mathbf{m}}_i, \boldsymbol{\theta}^m) \big) \frac{\partial}{\partial \theta_b^m} f_m(d, \hat{\mathbf{m}}_i, \boldsymbol{\theta}^m)$$

$$\frac{\partial}{\partial \theta_a^d} f_d(\hat{d}_i, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i, \boldsymbol{\theta}^d) = [\text{dcase}(\hat{d}_i, \hat{\mathbf{d}}_{\pi_i}, \hat{\mathbf{g}}_i) = a]$$

$$\frac{\partial}{\partial \theta_b^m} f_d(\hat{d}_i, \hat{\mathbf{m}}_i, \boldsymbol{\theta}^d) = [\text{mcase}(\hat{d}_i, \hat{\mathbf{m}}_i) = b]$$

Figure 8. Gradient Equations. The last two equations use Iverson's notation: $[X]$ is 1 is X is true, and 0 otherwise.

4.2. Learning

Following Eq. 17, the log likelihood for the training data $\{\hat{\mathbf{d}}, \hat{\mathbf{g}}, \hat{\mathbf{m}}\}$is

$$l = \sum_{\{\hat{\mathbf{d}},\hat{\mathbf{g}},\hat{\mathbf{m}}\}} \sum_i \big(f_d(\hat{d}_i, \hat{\mathbf{d}}_{\pi_i}, \mathbf{g}_i, \boldsymbol{\theta}^d) + f_m(d_i, \mathbf{m}_i, \boldsymbol{\theta}^m) - \log Z(\boldsymbol{\theta}^d, \boldsymbol{\theta}^m, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i) \big). \tag{19}$$

Even though it is discrete, the distribution $p(d|\mathbf{d}_\pi, \mathbf{g}, \mathbf{m})$ cannot be learned by counting. We again used a quasi-Newton method to optimize the log-likelihood. The gradients of the log likelihood are cumbersome to write down but easy to compute. These are shown in Fig. 8.

We used 100,000 samples from the Art, Books, Dolls, Laundry, Moebius, and Reindeer stereo pairs in the Middlebury dataset [16] for learning the parameters. The learning process took about 30 minutes using L-BFGS. If all parameters are fit together, we observed that the model tended to oversmooth the disparity map. Hence, we use two measures to reduce oversmoothing of disparity map. First, we train the matching parameters $\boldsymbol{\theta}^m$ by fixing $f_d = 0$, and then train the smoothness parameters with $\boldsymbol{\theta}^m$ held constant. Second, we regularize the smoothness term by changing the learnt $\boldsymbol{\theta}^d$ to $\boldsymbol{\theta}^d/\lambda$, where $\lambda = 1.5$.

4.3. Inference

For our final result, we would like to find the disparity map that has the smallest number of errors. The optimal solution to this is to take the disparity for *each pixel, independently,* to be the disparity that is most probable. To approximate this, we pursue the following strategy: sample several disparity maps from the posterior $p(\mathbf{d}|\mathbf{g}, \mathbf{m})$. Then, for each pixel, simply count the number of occurrences of each disparity, and set it to the most frequent. This is optimal in the sense that it minimizes the expected number of disparities that are incorrectly assigned. However, we emphasize that this is not equivalent to maximizing $p(\mathbf{d}|\mathbf{g}, \mathbf{m})$.

One major advantage of our directed model is that it is easy and efficient to obtain exact samples. It is possible to sample from our model in $O(Nd)$ where N is the number of pixels and d is the number of disparities. We emphasize that this is unlike the case for undirected models which in general require computationally intensive techniques such as Gibbs sampling. Sampling proceeds in a column major order. Notice, when proceeding in this order, that for each pixel, the parent disparities $\mathbf{d}_{\pi_i}$ in Fig. 7 will have already been sampled (The boundary cases are handled by setting disparity values at non-existent neighbors to the nearest available disparity. A more correct solution would be to learn different conditional distributions for the boundaries). The vector of matching costs $\mathbf{m}_i$, and the norm of the gradient $\mathbf{g}_i$, of course, are constant. Thus, we can easily sample d_i from $p(d_i|\mathbf{d}_{\pi_i}, g_i, \mathbf{m}_i)$. In our implementation, sampling one full disparity map from the Tsukuba pair (16 disparities) takes 0.1 seconds, and from the Cones pair (64 disparities) takes 0.4 seconds.

We compute 100 such samples of the disparity map. Half of the samples are obtained by flipping the stereo pair from left to right. These samples give us an approximate marginal distribution for the disparity at each pixel. For our final result, for each pixel, we choose the disparity that was sampled most often. Empirically, sampling 30 disparity maps gives a result close to the final solution. (We note that these samples could trivially be computed in parallel.)

We note that common inference algorithms, such as belief propagation, which attempt to maximize $p(\mathbf{d}|\mathbf{g}, \mathbf{m})$, are expensive to apply to our model, since it involves products over five disparity variables, each of which can take a significant number of values (16 to 80). However, taking the maximum probability disparity for each pixel will on average produce fewer errors than the joint maximum probability disparity map.

4.4. Stereo Experiments

Figure 9 and Table 3 show the computed disparity maps and scores for the Tsukuba, Venus, Teddy, and Cones stereo pairs from the Middlebury dataset. We compare against the CRF model of Scharstein and Pal[16] as well as the popular Graph Cuts[17] algorithm on which it is based. The scores indicate the percentage of pixels for which the absolute difference between the true disparity and computed disparity exceeded 1. We note that there are better performing hand crafted algorithms, but our results are similar to the model learnt using CRFs .

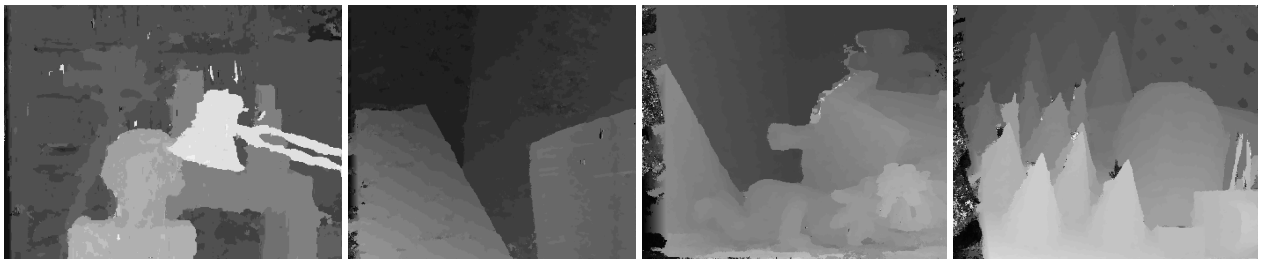

Figure 9. Stereo Results. From left to right: Tsukuba, Venus, Teddy, and Cones.

	Tsukuba	Venus	Teddy	Cones	Mean
Directed Sampling	3.9	3.6	10.5	4.2	5.6
Graph Cuts	1.9	1.8	16.5	7.7	7.0
CRFs ($K = 2$)	2.2	1.6	11.3	10.7	6.5

Table 3. Results of Middlebury evaluation. (Numbers indicate the percentage of incorrect disparities.)

5. Conclusions

Since directed models provide priors that seem to be comparable to the state of the art, and the fact that this kind of learning takes a few minutes, it suggests that directed models have great potential as a tool for computer vision research. Given the simplicity of our models here, we believe even better results can be obtained by learning more sophisticated representations.

6. Acknowledgements

The support of NSF under a collaborative project with Stanford University (EMT Bioinspired computing) is gratefully acknowledged. We also thank Daniel DeMenthon for helpful comments on a draft of this paper.

References

[1] `cs.ubc.ca/ schmidtm/Software/minFunc.html`.

[2] H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In *CVPR*, 2007.

[3] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, 2000.

[4] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Society B*, 36:192–225, 1974.

[5] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975).

[6] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *PAMI*, 20(4):401–406, 1998.

[7] D. DeMenthon, M. Vuilleumier, and D. Doermann. Hidden Markov Models for images. Technical report, Language and Media Processing Laboratory (LAMP), University of Maryland, 2000.

[8] A. J. Gray, J. W. Kay, and D. M. Titterington. An empirical study of the simulation of various models used for images. *PAMI*, 16(5):507–513, 1994.

[9] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006.

[10] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, March 2000.

[11] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook, Sept 2007.

[12] S. Roth. *High-Order Markov Random Fields for Low-Level Vision*. PhD thesis, Brown University, 2007.

[13] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

[14] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D depth reconstruction from a single still image. *IJCV*, 2007.

[15] A. Saxena, J. Schulte, and A. Y. Ng. Depth estimation using monocular and stereo cues. In *IJCAI*, 2007.

[16] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[18] M. Tappen, C. Liu, E. Adelson, and W. Freeman. Learning Gaussian conditional random fields for low-level vision. In *CVPR*, 2007.

[19] M. F. Tappen. Utilizing variational optimization to learn markov random fields. In *CVPR*, 2007.

[20] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *PAMI*, 27(9):1459–1472, 2005.

[21] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, November 1999.

[22] Y. Weiss and W. T. Freeman. What makes a good model of natural images? In *CVPR*, 2007.

[23] S. C. Zhu and D. Mumford. Prior learning and gibbs reaction-diffusion. *PAMI*, 19(11):1236–1250, 1997.