Learning for Stereo Vision Using the Structured Support Vector Machine

Yunpeng Li Daniel P. Huttenlocher
Department of Computer Science, Cornell University
Ithaca, NY 14853
{yuli,dph}@cs.cornell.edu

Abstract

We present a random field based model for stereo vision with explicit occlusion labeling in a probabilistic framework. The model employs non-parametric cost functions that can be learnt automatically using the structured support vector machine. The learning algorithm enables the training of models that are steered towards optimizing for a particular desired loss function, such as the metric used to evaluate the quality of the stereo labeling. Experimental results demonstrate that the performance of our method surpasses that of previous learning approaches and is comparable to the state-of-the-art for pixel-based stereo. Moreover, our method achieves good results even when trained on different image sets, in contrast with the common practice of hand tuning to specific benchmark images. In addition, we investigate the impact of graph structure on model performance. Our study shows that random field models with longer-range edges generally outperform the 4-connected grid and that this advantage is especially pronounced for noisy images.

1. Introduction

Stereo is among the most widely studied low-level problems in computer vision. It is an especially challenging task due to the inherent ambiguity in pixel matching, which is further complicated by phenomena such as occlusion and untextured regions. Random field models [3], which address the ambiguity problem by enforcing global consistency using spatial priors, have substantially advanced the state of the art of stereo vision, as noted in [21, 22]. Despite the progress, however, the parameters of most of these models remain hand-tuned. This is in sharp contrast with higher-level vision, such as object recognition, where machine learning is almost ubiquitous. Manually setting the parameters for low-level vision can be tedious, involving considerable human effort. This also limits our understanding of the adaptability of models, because of the difficulty of optimizing models by hand for new environments.

In this paper we present a conditional random field [13] based model for stereo vision with non-parametric cost functions, which can be learnt automatically using the structured support vector machine (structured SVM) [24] with linear kernels. We choose the discriminative conditional random field (CRF) over its generative counterpart, the Markov random field (MRF), because the former avoids the necessity to define a generative process, which is somewhat difficult to characterize in stereo. For instance it is common and often desirable to use gradient-adaptive spatial terms, which tend to violate the Markovian independence assumptions of generative models. Deviating from the traditional approach in random field based stereo, we use non-parametric cost functions to model the node and clique potentials of the CRF. In addition to providing flexibility in functional forms, this non-parametric approach allows us to express the total cost of the model as the inner product of a feature vector and a vector of the corresponding costs, where the costs correspond to the parameters of the model. The negated costs are commonly called "feature weights" in the machine learning literature. This linear form of the model enables us to use the structured SVM to learn the model parameters.[1]

The structured SVM is a large-margin method for estimating parameters, and can be an attractive alternative to the commonly used maximum likelihood estimator. A major advantage of the structured SVM, and large-margin methods in general, is that they take the loss function into consideration during training. Therefore, the approach can be used to train different models that specifically target different types of loss. In contrast, the maximum likelihood method is oblivious to loss; in fact it can be regarded as always minimizing the expected aggregate 0/1 loss (which is 0 if the labeling is *completely* correct and 1 otherwise). Such a loss function is clearly not optimal for most low-level vision problems, which usually have pixel-based performance criteria.

The main contribution of this paper is a formulation of stereo vision in terms of non-parametric CRF models and

[1] In this work, we only consider structured SVM with linear kernels.

a technique for training them using the structured SVM. This approach naturally allows us to learn models using the kinds of evaluation criteria that are normally used to assess stereo, such as the number of pixels whose labels are within 1 unit of the correct disparity level [16]. In our experiments we demonstrate that our method significantly outperforms other pixel-based stereo methods that have parametric (e.g. Potts) potentials trained using maximum likelihood. We also investigate the effect of the underlying graph structure on model performance, and show that the addition of explicit non-local interactions generally improves accuracy on more difficult scenes and especially in the presence of image noise.

1.1. Related Work

Random field models [3] have a long history of application in computer vision (e.g. [7, 20]). The classical Bayesian formulation decomposes the problem into a prior that enforces spatial consistency of the labels, and a likelihood function that encourages agreement between the labels and the observed data. For discriminative models, these two are more commonly called the *spatial term* and *data term* respectively. Inference on random fields with loops is generally intractable. However, high-quality approximate solutions are relatively easy to obtain owing to the development of efficient energy minimization methods, some of which are reviewed in [21, 22].

Commonly used forms of the spatial term are parametric functions of the disparity difference between neighboring pixels, which usually model the distribution as a mixture of a line process and an outlier process (e.g. in [19, 17, 29]). Common forms include Potts and truncated linear models. The functions are sometimes gradient adaptive (e.g. in [15]) to encourage discontinuity in disparity to coincide with change in image intensity. The data term is typically the value of some dissimilarity measure, such as the absolute intensity difference.

While these functional forms have been successfully used to produce good results, some fundamental issues remain unaddressed. Reasonable and intuitive as they are, parametric spatial terms such as the Potts and line-outlier models make particular assumptions about the form of the disparity distribution, which may not be true for the data. Therefore these models can be over-restrictive and fail to fit the data well. Using any dissimilarity measures directly as the cost function for the data term is also problematic. While a sophisticated metric, such as the sampling-insensitive dissimilarity [4], can provide a faithful measure of image difference and hence a reliable input to the data term, the metric *itself* is not necessarily a good cost function. In our model, the spatial term is a non-parametric function of disparity difference and discretized image gradient, and the data term is a non-parametric function of discretized dissimilarity value. While non-parametric stereo has been studied in many earlier works (e.g. [28, 2]), these approaches are typically based on ordering transforms and formulated as purely local methods rather than the global models that we investigate here.

Learning for stereo vision is a challenging subject. Considerable progress has been made in recent years, largely owing to the increasing availability of ground truth data. The work of [12] learns a probability model for matching errors using the scene structure of the input images. In [29], an expectation maximization (EM) algorithm is used to iteratively estimate disparity and re-learn the model parameters based on the estimate. While these methods have shown promising results, they do require some initial model whose parameters still need to be preset. Moreover they are conducted in a manner different from the standard settings of machine learning, where there are separate training and testing data. In these previous works, the model is learned from the same (unlabeled) data that is to be labeled, and the parameters are adjusted in order to improve performance. Our approach, on the other hand, learns the model from labeled training data and tests it on unseen inputs, which is a standard form of supervised training in machine learning.

A recent paper that employs this same supervised learning paradigm is [15], where a maximum likelihood estimator for the model parameters is obtained via gradient descent. Computing the likelihood gradient, however, involves the partition function, which is intractable on loopy graphs. In the aforementioned paper the partition function is approximated by the mode of the model distribution, which is obtained using graph cuts (GC) [5]. However the gradient tends to be noisy due to the approximation, as is observed in [15], which can lead to poor estimates.

Large margin methods are an alternative to the maximum-likelihood approach, and were originally introduced in the context of binary classification using optimal hyperplane separation [25]. The idea was first adapted to domains with structured output in the framework of max-margin Markov networks (M^3N) [23], where the required margin is rescaled by the loss of the inferred labeling. Since the set of linear constraints (of the M^3N quadratic program) is exponential in size, it is replaced with a non-linear constraint approximately solvable by linear programming relaxation. The method was subsequently applied to several low-level vision problems, including segmentation and terrain classification, demonstrating improvement [1] over the performance of previous models. Though a remarkable breakthrough, M^3N has its limitation. The linear programming formulation places a restriction on the form of admissible loss functions; more specifically, the per-label (i.e. per-pixel) loss function must be an indicator and must return zero if and only if the the inferred label is the exact same as the ground truth. In particular such a form of loss

function is not well suited to stereo, where the performance metric typically allows an error range around the true value (e.g., [16]).

The structured SVM [24] handles the exponential number of linear constraints in the quadratic program by employing a cutting-plane method. The algorithm iteratively finds the most violated constraint, i.e. the labeling with the smallest cost-less-loss value, and recomputes model parameters. The process is repeated until no significantly more violated constraint can be found. Thus the structured SVM places no restrictions on the form of loss functions, as long as the most violated constraint is feasible to compute under such loss. For random field based stereo, finding the exact most violated constraint is not tractable due the loopy graph structure; nonetheless an approximate one can be obtained using energy minimization techniques. In our work, we use loopy belief propagation (BP) [6, 14, 27] for this purpose and show that models trained with the approximate most violated constraints perform well in practice.

The rest of this paper is organized as follows. We define our model for stereo matching in Section 2. Section 3 describes the learning method in greater detail, and the experimental results are presented in Section 4. We conclude in Section 5.

2. CRF Model for Stereo

We model the problem of disparity labeling as a conditional random field on a grid graph. Hence each node, representing a pixel, is connected to its four nearest neighbors in both the horizontal and vertical direction. Later we will also formulate models with longer-range connections and investigate the impact on performance from the modified graph structure. For ease of presentation, however, we will start by describing the model defined on the conventional 4-connected grid.

Let $\mathcal{V}$ be the set of nodes and $\mathcal{E}$ be the set of edges in the graph. As is well known, the likelihood of a labeling X (i.e. the disparity map) given the input I decomposes into the product of maximal clique potentials and node potentials,

$$p(X|I;\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{(u,v)\in\mathcal{E}} \phi^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) \prod_{v\in\mathcal{V}} \phi^{\boldsymbol{\theta}}_{v}(x_v, I), \tag{1}$$

where $\boldsymbol{\theta}$ represents the parameters of the model and $Z(\boldsymbol{\theta})$ is the partition function. The notations $\mathbf{x}_{uv}$ and x_v denote the labeling over clique (u, v) and node v respectively. Note that the maximal cliques are simply the edges in the grid, since the graph is pairwise. As is a common practice, we assume that the distribution is in the general exponential family with $\phi^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) = \exp[-f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I)]$ and $\phi^{\boldsymbol{\theta}}_{v}(x_v, I) = \exp[-g^{\boldsymbol{\theta}}_{v}(x_v, I)]$, where $f^{\boldsymbol{\theta}}_{uv}$ and $g^{\boldsymbol{\theta}}_{v}$ are cost functions for the spatial and the data terms respectively. Hence the *cost of labeling* X, given input I, can be defined in the negative log-likelihood space as

$$\begin{aligned} E_{\boldsymbol{\theta}}(X, I) &= -\log p(X|I;\boldsymbol{\theta}) - \log Z(\boldsymbol{\theta}) \\ &= \sum_{(u,v)\in\mathcal{E}} f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) + \sum_{v\in\mathcal{V}} g^{\boldsymbol{\theta}}_{v}(x_v, I). \end{aligned} \tag{2}$$

This quantity is also commonly referred to as the *energy* of the random field, and we will use the words "energy" and "cost" interchangeably. Since the input I is a constant and the log partition function $\log Z(\boldsymbol{\theta})$ does not depend on X, finding a labeling that maximizes the likelihood $p(X|I;\boldsymbol{\theta})$ is equivalent to finding one that minimizes the cost $E_{\boldsymbol{\theta}}(X, I)$.

The input for stereo consists of two images $I = (I_L, I_R)$, where I_L is the one taken by the left camera and I_R by the right. We assume without loss of generality that the disparity map is always computed for the left camera scene I_L. Since we model occlusion explicitly, the set of labels include the set of integer disparity levels plus occlusion.

2.1. Spatial Term

The spatial cost $f^{\boldsymbol{\theta}}_{uv}$ is a function of disparity levels at neighboring pixels u and v as well as the local image gradient. More specifically,

$$f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) = f^{\boldsymbol{\theta}}_{uv}(J(x_u, x_v), K(u, v)), \tag{3}$$

(recall from Equation 1 that $\mathbf{x}_{uv}$ and x_v are the labelings of the clique (u, v) and the node v respectively) with discrete valued functions J and K

$$J(x_u, x_v) = \begin{cases} x_v - x_u & \text{if neither } u \text{ nor } v \text{ is occluded} \\ left\_occl & \text{if } u \text{ is occluded} \\ right\_occl & \text{if } v \text{ is occluded} \\ 0 & \text{if both } u \text{ and } v \text{ are occluded} \end{cases} \tag{4}$$

and

$$K(u, v) = \lfloor |I'_L(v) - I'_L(u)| \rfloor. \tag{5}$$

For $f^{\boldsymbol{\theta}}_{uv}$ we assume that (u, v) is in the horizontal direction, since the case for vertical direction is entirely analogous. In $K(u, v)$, I'_L is I_L after a small amount of Gaussian smoothing, which is applied to reduce the impact of texture and noise. In the case of color images, $|I'_L(v) - I'_L(u)|$ is averaged over the color channels.

Since the structured SVM requires the model to have a linear discriminative function, the cost function $f^{\boldsymbol{\theta}}_{uv}(J(x_u, x_v), K(u, v))$, abbreviated as $f^{\boldsymbol{\theta}}(J, K)$, has to be linear; in other words, $f^{\boldsymbol{\theta}}(J, K)$ needs to be expressible as the inner product of a parameter vector and some feature vector. This can be achieved using the following form for the cost function

$$f^{\boldsymbol{\theta}}(J, K) = \theta_{f(jk)} \qquad \text{if } J = j \text{ and } K = k \ , \tag{6}$$

where $\theta_{f(jk)}$ are real valued model parameters. Let $\psi_{uv}(j,k)$ be the indicator function that $J(x_u, x_v) = j$ and $K(u,v) = k$, i.e. it is one if the condition holds and zero otherwise. Let $\boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I)$ denote the vector whose entries are $\psi_{uv}(j,k)$ for each combination of j and k at clique (u,v). Let $\boldsymbol{\theta}_f$ be the vector that contains the corresponding parameters $\theta_{f(jk)}$. Hence $f^{\boldsymbol{\theta}}_{uv}$ can be written as the inner product of $\boldsymbol{\theta}_f$ and $\boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I)$, i.e.

$$f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) = \langle \boldsymbol{\theta}_f, \boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I) \rangle \,. \tag{7}$$

For notational convenience we assume that horizontal and vertical cliques (i.e. edges in pairwise models) share the same spatial parameters. Below we will discuss the extension to anisotropic clique potentials, which is straightforward. We define the *spatial feature vector*

$$\boldsymbol{\Psi}_f(X, I) = \sum_{(u,v)\in\mathcal{E}} \boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I). \tag{8}$$

Hence the total spatial cost (i.e. the first term of Equation 2) is

$$E_{\boldsymbol{\theta},f}(X, I) = \langle \boldsymbol{\theta}_f, \boldsymbol{\Psi}_f(X, I) \rangle \,. \tag{9}$$

When horizontal and vertical cliques have different potentials, we simply have separate parameter and feature vectors for each type of cliques. The overall vectors are just the concatenations over the different clique types, and hence the cost is still the inner product of the parameter vector and the feature vector as in Equation 9. The same extension also applies directly to the scenario where there are multiple classes of edges in the graph that may correspond to various lengths and orientations.

2.2. Data Term

Similar to the spatial cost, the data cost $g^{\boldsymbol{\theta}}_v$ is also defined as a non-parametric function

$$g^{\boldsymbol{\theta}}_v(x_v, I) = \begin{cases} c_v \cdot \theta_{g(k)} & \text{if } v \text{ is not occluded and} \\ & \lfloor \delta(v, I_L, v - x_v, I_R) \rfloor = k \\ c_v \cdot \theta_{g(occl)} & \text{if } v \text{ is occluded} \end{cases} \tag{10}$$

where c_v is some constant scalar and $\delta(v, I_L, v - x_v, I_R)$ is the sampling-insensitive dissimilarity [4] between pixel v in image I_L and its match in I_R.

As before, let $\boldsymbol{\theta}_g$ be the vector containing all data term parameters $\theta_{g(k)}$ (including $\theta_{g(occl)}$) and let $\boldsymbol{\psi}_v(x_v, I)$ be the corresponding vector of indicators for the conditions in Equation 10. Hence $g^{\boldsymbol{\theta}}_v$ is the inner product of $\boldsymbol{\theta}_g$ and $c_v\boldsymbol{\psi}_v(x_v, I)$

$$g^{\boldsymbol{\theta}}_v(x_v, I) = \langle \boldsymbol{\theta}_g, c_v\boldsymbol{\psi}_v(x_v, I) \rangle \,. \tag{11}$$

Analogous to spatial features, the *data feature vector* is defined as

$$\boldsymbol{\Psi}_g(X, I) = \sum_{v\in\mathcal{V}} c_v\boldsymbol{\psi}_v(x_v, I). \tag{12}$$

We let c_v equal the degree of node v, so that the ratio between the total counts of spatial features and data features is constant with respect to the number edges. This prevents potential imbalance between the norms of the spatial and the data feature vectors when the model has multiple families edges and hence a higher edge-to-node ratio. Thus it ensures that SVM never places too much attention on one type of features and not enough on the other.

The total data cost (i.e. the second term of Equation 9) is also the inner product of the data parameter vector and the data feature vector,

$$E_{\boldsymbol{\theta},g}(X, I) = \langle \boldsymbol{\theta}_g, \boldsymbol{\Psi}_g(X, I) \rangle \,. \tag{13}$$

Therefore, the total cost of labeling X given input I is

$$E_{\boldsymbol{\theta}}(X, I) = \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(X, I) \rangle \tag{14}$$

where parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_f^T, \boldsymbol{\theta}_g^T)^T$ and feature vector $\boldsymbol{\Psi}(X, I) = (\boldsymbol{\Psi}_f(X, I)^T, \boldsymbol{\Psi}_g(X, I)^T)^T$ are both concatenated over the spatial and the data terms. The desired labeling under the model is simply the one with minimum cost.

2.3. Graph Structure with Long-range Edges

In addition to the grid graph, we also explore structures with long-range edges. In particular, we consider horizontal and vertical edges that have length 3^k for $k = 0, 1, 2, \cdots, K-1$. The larger K, the greater the maximum range of explicit interaction is modeled. Thus the grid graph is a special case where $K = 1$. We choose 3 as the base, since it is the smallest integer for which the random field remains strictly pairwise (i.e. the maximal cliques are still of size two and thus the same formalization applies). The exponentially increasing edge length also enables us to model longer range of interaction at relatively lower computational expense, compared with earlier models with denser edges (e.g. [8]).

The cost function in this more general setting is still the inner product between parameters and features, where the spatial term vectors are concatenated over each type of edges. Hence the form of Equation 14 remains valid under this extension.

3. Parameter Learning

The model parameters are learnt using the structured SVM [24]. Let $((I^{(1)}, X^{(1)}), \cdots, (I^{(n)}, X^{(n)}))$ be the training examples, each of which is an input-output pair. The structured SVM optimizes for parameters $\boldsymbol{\theta}$ by minimizing a quadratic objective function subject to a set of

linear soft margin constraints

$$\min_{\boldsymbol{\theta},\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{\theta}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \quad (15)$$
$$\text{s.t. } \forall i, \forall X \in \mathcal{X} : \langle \boldsymbol{\theta}, \boldsymbol{\delta\Psi}_i(X)\rangle \geq \Delta(X^{(i)}, X) - \xi_i$$

where $\mathcal{X}$ is the set of all possible labelings, ξ_i are the slack variables associated with each example, and $\Delta(X^{(i)}, X)$ is the loss function, which we will define later in this section. Also $\boldsymbol{\delta\Psi}_i(X)$ denotes $\boldsymbol{\Psi}_i(X) - \boldsymbol{\Psi}_i(X^{(i)})$ with $\boldsymbol{\Psi}_i(X)$ being shorthand for $\boldsymbol{\Psi}(X, I^{(i)})$, and $C > 0$ is a constant that controls the trade-off between margin and training error. Rearranging the terms of Equation 16 shows that the SVM objective function is an upper bound on average training loss (up to a constant factor C), as long as a labeling with cost no higher than that of the ground truth can be found for every training example. While this condition is not guaranteed due to the intractability of exact energy minimization on loopy graphs, it is often true in many real-world low-level vision problems and especially stereo [21].

The apparent difficulty in this formulation is the exponential sized labeling set $\mathcal{X}$. The structured SVM addresses this problem by replacing it with a collection of finite constraint sets S_i. Initially all the constraint sets S_i are empty and the parameter vector $\boldsymbol{\theta}$ is set to some arbitrary value, typically all-zeros. At each iteration and for each example i, the algorithm computes the most violated constraint, i.e. one with the largest slack ξ_i, and adds it to the constraint set S_i if it is more violated than those already in the set. The solution to the quadratic program is then recomputed and hence $\boldsymbol{\theta}$ updated. The algorithm iterates until no new constraints are added.

Since maximizing ξ_i is equivalent to minimizing $\langle \boldsymbol{\theta}, \boldsymbol{\delta\Psi}_i(X)\rangle - \Delta(X^{(i)}, X)$ and $E_{\boldsymbol{\theta}}(X^{(i)}, I^{(i)}) = \langle \boldsymbol{\theta}, \boldsymbol{\Psi}_i(X^{(i)})\rangle$ is a constant, the most violated constraint for example i is just the one with the smallest cost-less-loss value

$$\hat{X} = \arg\min_{X\in\mathcal{X}} \Big\{ E_{\boldsymbol{\theta}}(X, I^{(i)}) - \Delta(X^{(i)}, X) \Big\}. \quad (16)$$

For any per-pixel loss function, approximate solutions for $\hat{X}$ can be obtained efficiently using energy minimization techniques. It is worth noting that the structured SVM also provides several other formulations of the quadratic program [24]. However, the version with linear slack penalties and margin rescaling (Equation 16) is the only one under which there are known efficient approximation algorithms for $\hat{X}$ in stereo.

A challenge for learning non-parametric functions using the structured SVM is that the parameters, namely the discrete cost function outputs, are treated as independent variables by the learning algorithm, and hence the learnt cost functions may have certain characteristics that are unnatural for the underlying problem. In stereo, this is mainly manifested as fluctuations in the shape of the data cost function. Though the learnt function does have the expected overall trend of increasing with dissimilarity, it is not strictly monotone as it should be for stereo. We address this problem by imposing a monotonicity constraint on the data cost function after training. This is done by setting $\theta_{g(k)}$ to $\min(\theta_{g(k)}, \theta_{g(k+1)})$ in decreasing order of k ($k \neq occl$, i.e. occlusion cost is unchanged). In this way, we capture the domain-specific knowledge without further restricting the form of the cost function.

We also noticed that the lowest training error is usually achieved not by the final output of the SVM, but by some $\boldsymbol{\theta}$ produced after one of the intermediate training iterations. This is not surprising since the SVM objective function is not the same as training error, even though it is an upper bound on the loss. One reason to formulate learning as constrained optimization of such a bound is that directly minimizing training error is usually not feasible. Also in SVM theory, minimizing the norm of the learnt parameter vector (i.e. the first term of the objective function) is equivalent to increasing the margin [25] and hence guards against overfitting. For our stereo learning problem, however, we found that overfitting hardly occurs and that generalization error is much more closely correlated with training error than with the value of the SVM objection function. Therefore, we choose from all learnt $\boldsymbol{\theta}$ vectors (produced after each iteration) the one with the lowest training error as the model parameter. Parameters learnt in this way are still large margin estimates since they are obtained through SVM optimization. This modified training procedure can be considered as exploring a subset of the parameter space that has the large margin properties, and choosing the best instance based on training performance.

3.1. Loss Functions

The most natural choice of loss function is simply the error function under which model performance is evaluated. For stereo this is usually the number of bad pixels in non-occluded regions (determined by the ground truth), where a pixel is bad if the disparity estimated by the model differs from the true disparity by an amount greater than some threshold r. The conventional choice in stereo is $r = 1$, which we use in our work. Hence the loss function is

$$\Delta(l) = \sum_{v\in\mathcal{V}} l(v) \quad (17)$$

where l is the pixel-wise loss and in the case of standard stereo evaluation metric it is

$$l_{std}(v) = \begin{cases} 1 & \text{if } v \text{ is bad and not in occluded regions} \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Both l and $\Delta(l)$ take as arguments the ground truth and the proposed labeling, which are omitted from the notation above for conciseness. It is easy to see that l_{std} discourages labeling of occlusion, since every non-occluded pixel mislabeled as occlusion encounters a loss while there is no penalty for occluded pixels mislabeled as non-occlusion. Such a loss function tends to produce models that label very little or no occlusion, though this is consistent with the goal of achieving the best performance in non-occluded regions.

We can extend the definition of bad (i.e. mislabeled) pixel to occluded region. In particular, we consider a pixel a *false negative* if it is occluded in the ground truth but not labeled so by the model; similarly it is a *false positive* if the opposite happens. In either case, the pixel is regarded as mislabeled. We can define a new pixel-wise loss that is aimed at achieving lower overall error rates by correctly identifying occluded regions

$$l_{occl}(v) = \begin{cases} q & \text{if } v \text{ is a false positive} \\ 1 & \text{if } v \text{ is otherwise mislabeled} \\ 0 & \text{if } v \text{ is correctly labeled} \end{cases} \quad (19)$$

where constant q adjusts the extent to which occlusion labeling shall be encouraged. If $q = 1$ then $\Delta(l_{occl})$ would measure the model performance as the number of bad pixels over the whole scene. For use as an SVM loss function, we find that a smaller value of 0.06 proves to be a better choice for improving overall accuracy.

4. Experimental Results

For performance evaluation we train and test our model mainly on the Middlebury-2005 stereo data set release in [15], which contains scenes that are more complex and challenging than the older ones on the Middlebury Stereo Evaluation page [16]. Since the stereo benchmark does not label occlusions, we simply fill in the occluded region by replacing the occluded pixel (inferred by the model) with the disparity of the first non-occluded pixel to its left (or to its right if it is near the left boundary) when evaluating performance. This is obviously suboptimal and fails to exploit the full benefit of occlusion labeling; nonetheless, finding a good extrapolation scheme for occluded regions is beyond the scope of this work. For training of all models, we use Joachims's *SVM-struct* [24] with a C value (see equation 16) of 10^{-3} that is empirically chosen based on training error. The outcomes are nevertheless rather insensitive to the choice of C, and in fact values from 10^{-4} to 10^{-1} produce models that are indistinguishable in performance.

We compare our results with several other pixel-based stereo algorithms [15, 16, 18], and show that our model achieves a high level of performance. The error rates of our models are compared with those of [15] whenever possible (i.e. when the corresponding data is available in [15]). The comparison with [16] and [18], both non-learning based, is limited to the "Teddy" and "Cones" scenes, since these algorithms predate [15] and hence no results are reported on the Middlebury-2005 data. It should be noted that our method is raw-pixel based and treats stereo as a generic random field labeling problem, and does not use techniques such as local support windows, segmentation, or plane fitting (e.g. [11, 26, 30]). Therefore comparison with these more specialized stereo algorithms is not meaningful. However, many of these more involved methods use MRF or CRF models at some stage, and thus our learning technique should prove useful to further work on such approaches to stereo.

Figure 1. Sample disparity maps for stereo scenes Art and Cones produced by long-range CRF models ($K = 3$) learnt with l_{occl} loss function. For "Art" (top), the model is trained on the rest 5 scenes in the same data set; for "Cones" (bottom), the model is trained on all 6 scenes in the Middlebury-2005 data set (which contains "Art" but not "Cones"). Occluded regions inferred by the model are masked in full black.

The results in Table 1 show that our method achieves performance superior to that of [15], which also uses machine learning. Comparing with the non-learning based methods, the performance of our learnt models by far surpasses [16] and is comparable with [18], one of the top-performing stereo algorithms. This is despite the fact that [18] generates a second disparity map using the other image of the stereo pair in order to exploit visibility constraints, while our models are generic random fields and do not make use of this property.

Table 2 shows the error rates of leave-one-out cross validation, where for each scene the model is trained on all the other scenes in the data set. In addition we also train our model on the 2006 data set from the Middlebury Stereo website, which has very different characteristics, and test it on the 2005 data set. As one can see, the performance of leave-one-out cross validation is close to that of training on

Model \ Scene	Art	Books	Dolls	Laundry	Moebius	Reindeer	**average**	Teddy	Cones
- Grid ($K = 1$), l_{std} loss	14.66	19.12	12.70	19.16	10.88	**11.72**	14.71	11.34	4.68
- Grid, l_{occl}	15.24	21.13	**12.11**	17.14	11.28	16.47	15.56	10.92	4.27
- Long-range ($K = 3$), l_{std}	**12.11**	**15.68**	12.14	15.82	**10.80**	15.26	**13.64**	8.89	3.94
- Long-range, l_{occl}	12.69	16.29	12.57	**15.79**	11.30	15.70	14.06	8.15	**3.77**
- [15] w/ 2 gradient bins	–	–	–	–	–	–	18†	11.3	10.7
- [15] w/ 6 gradient bins	–	–	–	–	–	–	20	14.5	16.8
- [16] w/ GC (non-learning)	–	–	–	–	–	–	–	16.5	7.70
- [18] (non-learning)	–	–	–	–	–	–	–	**6.47**	4.79

Table 1. Performance of models on the Middlebury-2005 data set [15] as well as the "Teddy" and "Cones" scenes from the Middlebury Stereo Evaluation page [16]. Here the learnt models are trained on all the 6 scenes in Middlebury-2005. The table shows error rates measured as the percentage of bad pixels, lower is better, calculated in non-occluded regions (as is common). Bold fonts indicate the lowest error rates among the models being compared, and "–" indicates result not available.

† Read from the plots in Figure 6 – 8 of [15].

Model \ Scene	Art	Books	Dolls	Laundry	Moebius	Reindeer	**average**
Leave-one-out							
- Grid, l_{std}	15.54	20.81	12.83	18.21	11.69	**13.04**	15.35
- Grid, l_{occl}	15.11	21.97	12.88	18.10	11.13	14.09	15.55
- Long-range, l_{std}	**12.77**	17.56	**12.40**	**16.75**	11.25	15.41	**14.36**
- Long-range, l_{occl}	13.49	**15.98**	12.89	17.06	**10.64**	18.94	14.83
- [15] w/ 2 gradient bins	–	–	–	–	17	14	–
- [15] w/ 6 gradient bins	–	–	–	–	13	18	–
Train on Middlebury-2006							
- Long-range, l_{std}	**13.60**	**16.13**	13.86	20.65	**12.21**	17.90	15.73
- Long-range, l_{occl}	14.37	16.79	**12.87**	**17.14**	12.84	**15.76**	**14.96**

Table 2. Performance of learnt models in leave-one-out cross validation on the Middlebury-2005 data set (top) and performance of models trained on Middlebury-2006 and tested on Middlebury-2005 (bottom). The error rates are measured in the same way as in Table 1.

the whole data set; moreover, training on a very different data set yields only slightly higher error rates. This indicates that model and the training method generalize well to unseen data.

Another observation from the two tables is that when the models are trained directly using loss functions that encourage occlusion labeling (i.e. l_{occl}) they have nearly the same level of performance as those trained using the evaluation metric itself as the loss function (i.e. l_{std}). This demonstrates that our method is able to handle occlusion without sacrificing much accuracy in the non-occluded regions. Figure 1 displays some sample output disparity maps produced by models trained with l_{occl} loss. One can see that most of the occluded regions are correctly identified.

Moreover, the figures in both Table 1 and 2 suggest that models with explicit long-range interactions generally perform better than those with only local connections, namely the grid model (e.g. compare row 1, 2 with row 3, 4). This indicates that the inclusion of sparse long-range edges does yield some benefit. To investigate this further, we study the trends in which model performance degenerates with increasing image noise. This bears practical concern for stereo, since in real-world situations the input images are unlikely to be as noise-free as those taken in the lab. In fact, noise in stereo has been a subject of study in several recent papers, e.g. [10, 9].

Model \ Noise σ	3	5	7	10
Grid	18.84	24.18	32.25	46.44
Long-range	15.55	18.23	21.20	24.20

Table 3. Model performance on noisy stereo input (Middlebury-2005 data set). The images from the testing scenes are corrupted by additive Gaussian noise with standard deviation σ. Models are trained on the original (non-noisy) images with l_{std} loss. (The scenario for using l_{occl} is essentially the same.) Evaluation is based on leave-one-out cross validation and the error rates are averaged over the whole data set.

Table 3 shows the percentage error rates of grid and long-range models on stereo inputs with Gaussian noise. Here the difference is much more pronounced. The performance of the 4-connected grid model rapidly declines as the noise level increases, whereas the one with long range connec-

tions undergoes a much more graceful degradation. Note that the goal of this comparison is not to develop a new method for noisy stereo, which is itself a separate research topic; it is simply shows the advantage of increased robustness of long-range models over the grid under equal conditions.

5. Conclusion

We presented a technique for learning random field based non-parametric models for stereo using the structured support vector machine. Experiments illustrate that our method achieves significantly better performance than previous learning approaches and moreover is capable of explicitly labeling occlusion. We also found that models with long-range interactions generally outperform the grid model, which has only local connections; the performance gap becomes more evident as the noise level increases. Though only applied to stereo, our model is formulated as a generic random field labeling problem and the learning algorithm makes few assumptions specific to stereo. As such, it can be adapted to other low-level vision problems and hence may serve as a useful tool for other research areas.

Acknowledgments

This work was supported in part by NSF grant IIS-0713185. The authors would like to thank Thorsten Joachims for helpful discussions.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *CVPR*, 2005.

[2] J. Banks, M. Bennamoun, and P. Corke. Non-parametric techniques for fast and robust stereo matching. In *TENCON*, 1997.

[3] J. E. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc., B*, 36(2), 1974.

[4] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, 1998.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, 1999.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), 2006.

[7] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6(6):721–741, November 1984.

[8] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Intl. Congress of Mathematicians*, 1986.

[9] Y. S. Heo, K. M. Lee, and S. U. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *CVPR*, 2007.

[10] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007.

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

[12] D. Kong and H. Tao. A method for learning matching errors in stereo computation. In *BMCV*, 2004.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[14] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

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

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

[17] C. Strecha, R. Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, pages 2394–2401, Washington, DC, USA, 2006. IEEE Computer Society.

[18] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[19] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *PAMI*, 25(7), 2003.

[20] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *IJCV*, 5(3), 1990.

[21] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *ECCV*, 2006.

[22] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, 2003.

[23] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[24] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[25] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[26] Q. Yáng, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, 2006.

[27] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2000.

[28] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994.

[29] L. Zhang and S. M. Seitz. Parameter estimation for MRF stereo. In *CVPR*, 2005.

[30] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1):49–65, 2007.