# Learning the Right Model: Efficient Max-Margin Learning in Laplacian CRFs

Dhruv Batra
TTI-Chicago
dbatra@ttic.edu

Ashutosh Saxena
Cornell University
asaxena@cs.cornell.edu

## Abstract

*An important modeling decision made while designing Conditional Random Fields (CRFs) is the choice of the potential functions over the cliques of variables. Laplacian potentials are useful because they are robust potentials and match image statistics better than Gaussians. Moreover, energies with Laplacian terms remain convex, which simplifies inference. This makes Laplacian potentials an ideal modeling choice for some applications.*

*In this paper, we study max-margin parameter learning in CRFs with Laplacian potentials (LCRFs). We first show that structured hinge-loss [35] is non-convex for LCRFs and thus techniques used by previous works are not applicable. We then present the* first *approximate max-margin algorithm for LCRFs. Finally, we make our learning algorithm scalable in the number of training images by using dual-decomposition techniques. Our experiments on single-image depth estimation show that even with simple features, our approach achieves comparable to state-of-art results.*

## 1. Introduction

Undirected graphical models such as Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) have been successfully applied to a number of vision problems, such as image denoising, optical flow and single-image depth estimation. While designing an MRF/CRF for an application, especially one with continuous random variables, an important modeling decision is the choice of the family of potential functions over the cliques of variables.

In the context of natural images, this question has been studied as the search for suitable natural image priors [36, 38]. Some of the earliest works [12] used quadratic disagreement pairwise potentials, corresponding to Gaussian priors on images. Since then however, a large body of work [21, 34, 36, 38] has found that histograms of filter responses for natural images tend to be highly "non-Gaussian", in that they have sharp peaks at zero and heavy tails. Consequently, recent works have focused on non-convex priors [2, 22, 23, 32, 36].

A similar situation holds for *range images*, *i.e.* images captured by laser range-scanners as opposed to traditional cameras. Huang *et al.* [13] presented the first analysis of range images and found that log-gradient-histograms of range images of natural scenes were also heavy-tailed and peaked at zero. More recently, Saxena *et al.* [24, 25] made similar observations in the context of monocular depth estimation, and found that relative depths are better modeled by a Laplacian distribution than a Gaussian.

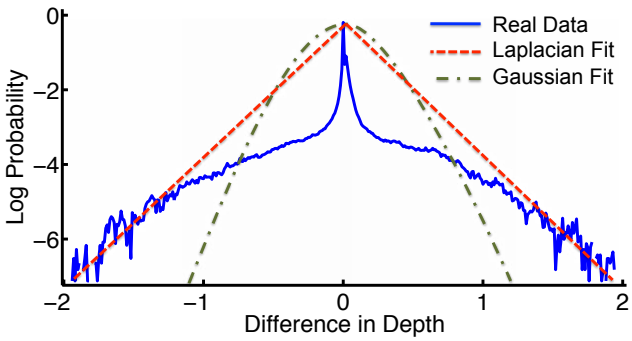



Figure 1: Log10 of the normalized histogram of relative depths (between adjacent pixels) from 400 laser scans collected by Saxena *et al.* [24, 25]. Notice that the relative depths are better modeled by a Laplacian distribution than a Gaussian.

**Model.** The model we consider is a CRF with Laplacian potentials, which we refer to as Laplacian CRF (LCRF) for ease of notation. Although non-convex models like Fields of Experts (FOE) [22, 36] or hyper-Laplacian priors [14] may be a better fit to natural statistics than LCRFs, there are a number of good reasons for using LCRFs.

Laplacian potentials represent a sweet spot in the trade-off between the conflicting goals of modeling and optimization. Gaussian potentials lead to easy (inference and learning) optimization problems, but are a poor match to image statistics. Non-convex models (e.g., FOE) match image statistics well but result in difficult (non-convex) optimization problems. Laplacian potentials are robust potentials and match image statistics better than Gaussians, yet energies with Laplacian terms remain convex, which simplifies inference. Moreover, in recent work, Schmidt *et al.* [29] found that Laplacian models actually outperformed hyper-Laplacian models on the task of image restoration, when

used with MAP inference.

**Goal.** In this paper, we study discriminative parameter learning in LCRFs. This is a challenging problem because LCRFs involve $\ell_1$-norm terms and thus the energy function (negative log probability) is a non-linear function of the parameters. Thus, well understood techniques like Structured SVMs (SSVMs) [35] and Max-Margin Markov Nets ($M^3$Ns) [5] are not directly applicable.

**Contributions.** We first show that the key object in max-margin learning, *i.e*. the structured hinge-loss [35] is non-convex for LCRFs. Thus, an exact max-margin learning algorithm is unlikely to exist. We then present an *approximate* max-margin algorithm for LCRFs by linearizing the non-convex $\ell_1$-norm constraints. This broadens the class of energy functions that may be learnt via SSVMs, albeit approximately. To the best of our knowledge, this is the first max-margin discriminative training algorithm for CRFs with Laplacian potentials.

In addition, we use ideas from the dual-decomposition [3, 7] literature to decompose the problem of learning parameters from a dataset of images into smaller learning problems over individual training images. We present an efficient dual-decomposition-based algorithm that scales linearly with the number of training images and is very efficient in practice. This makes our approach highly parallelizable and scalable to a large number of training images.

We apply LCRFs to the problem of single-image depth estimation, which is a difficult mathematically-ill-posed problem due to the ambiguities introduced by the projection of the 3D world onto a 2D image. Interestingly, for this problem, Saxena *et al*. [24] originally proposed an LCRF to model depth as a function of the image features. However, in the absence of a parameter learning algorithm, they resorted to a heuristic approach that neglected the partition function. In this work, we show that by using a principled approximate learning algorithm, we obtain improvements in depth estimates, not only over their heuristic approach but also other techniques. Specifically, we achieve state-of-art performance on one common error metric and are competitive with the state of the art on another metric.

## 2. Related Work

Most relevant to our work are algorithms for parameter learning in continuous random field models, max-margin methods and techniques for single-image depth estimation.

**Parameter Learning in Continuous Random Fields.** In the FOE model, Roth and Black [22] used contrastive divergence [10] to approximate the maximum likelihood estimation of the parameters. Weiss and Freeman [36] proposed a basis rotation algorithm for approximating the same. Note that these are generative training methods while we are interested in a discriminative training algorithm.

Tappen *et al*. [33] presented a Gaussian CRF model, and showed that discriminative learning in GCRFs boils down to linear algebra operations, and is thus tractable and efficient. Scharstein and Pal [28] used MAP estimates to approximate the gradient for max-conditional-likelihood learning of parameters. In [32], Tappen trained FOE parameters by minimizing a loss function with stochastic gradient descent. Samuel and Tappen [23] presented an improved version based on implicit-differentiation. Li and Huttenlocher [18] presented a discriminative learning algorithm based on simultaneous perturbation stochastic approximation. Barbu [2] used marginal space learning to learn the parameters. Note that for LCRFs, any reasonable loss function will be non-differentiable due to $\ell_1$-norm terms, and thus gradient-based methods are not directly applicable. Instead of exploring smooth approximations (which are often slow to converge), we formulate our problem as a Structured SVM (solved via a cutting-plane algorithm).

**Structured Max-Margin Learning.** Taskar *et al*. [5] proposed a max-margin method for training Markov networks and Tsochantaridis *et al*. [35] proposed a Structured Support Vector Machine (SSVM) framework for learning structured-output models. Both techniques have been widely used since their introduction. Li and Huttenlocher [19] proposed an SSVM-based algorithm in the context of stereo. Szummer *et al*. [31] used graph-cuts within an SSVM learning algorithm in the context of segmentation. We note that in both cases, the energies of the models were linear in the parameters. This is not true for LCRFs, and thus max-margin methods are not directly applicable. Overcoming this restriction is the main focus of this paper.

**Single Image Depth Estimation.** Saxena *et al*. [24, 27] considered the problem of depth estimation from a single image using a CRF. They found that CRFs with Laplacian potentials significantly outperform those with Gaussian potentials, even with their heuristic learning approach, in which they ignored the partition function. Sudderth *et al*. [30] used hierarchical Dirichlet Processes in order to model the depth of objects. Liu *et al*. [20] proposed a semantic-category based depth-estimation model that is the current state-of-art (in terms of one error metric) on the dataset of Saxena *et al*. [24,27]. More recently, Li *et al*. [16] proposed a feedback-enabled cascaded classification model that achieved state-of-art performance (in terms of another error metric) on this dataset.

**Laplacian terms.** We note that Laplacian terms have been explored in several contexts, including Lasso shrinkage [8] and sparse coding [6]. These methods focus on *inference* techniques, *i.e*. how to efficiently minimize objective functions with $\ell_1$-norm terms or constraints. This paper, on the other hand, is concerned with the problem of parameter *learning* in CRFs that contain these potentials.

Finally, we should also point out that although the work of Zhu *et al*. [37] uses a similar name as us, they use Laplacian priors for a sparse structural bias, while we use Laplacian potentials on the variables of the model.

## 3. Laplacian CRF

We now describe our model in detail before presenting our proposed learning algorithm in Section 4.

**Notation.** Let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. Consider a collection of continuous random variables $\mathcal{Y} = \{y_i \mid i \in [n], y_i \in \mathbb{R}\}$, and a graph $G = (\mathcal{V}, \mathcal{E})$ defined over these variables, *i.e*. $\mathcal{V} = [n]$, $\mathcal{E} \subseteq \binom{[n]}{2}$. For a vector $\boldsymbol{y} \in \mathbb{R}^n$, we use $P(\boldsymbol{y})$ as a shorthand for $P(\mathcal{Y} = \boldsymbol{y})$. Our goal is to jointly predict $\mathcal{Y}$ from a collection of local features $\{\boldsymbol{x}_i \in \mathbb{R}^k \mid i \in [n]\}$ extracted at these labeling sites. Let $\mathcal{X} = [\boldsymbol{x}_1^T; \boldsymbol{x}_2^T; \ldots; \boldsymbol{x}_n^T]$ be the matrix holding these features as rows. Finally, let $Q = [\boldsymbol{f}_1^T; \ldots; \boldsymbol{f}_m^T]$ be a matrix of linear filters operating on $\mathcal{Y}$.

We define a Laplacian CRF as:

$$P(\boldsymbol{y} \mid \mathcal{X}, \theta) = \frac{1}{\mathcal{Z}} \exp\left(-E(y|\mathcal{X}, \theta)\right), \quad \text{where} \tag{1}$$

$$E(\boldsymbol{y} \mid \mathcal{X}, \theta) = ||\boldsymbol{y} - \mathcal{X}\theta||_1 + ||Q\boldsymbol{y}||_1, \tag{2}$$

and where $||\boldsymbol{a}||_1 = \sum_{i=1}^m |a_i|$ for $\boldsymbol{a} \in \mathbb{R}^m$, $\mathcal{Z}$ is the partition function, and $\theta \in \mathbb{R}^k$ is the vector of model parameters (to be learnt). We can see that this model penalizes for deviations from linear predictions and for having large responses to the filters $Q$. Comparing our model with popular models like [32], we note that they penalize filter responses via a non-convex Lorentzian penalty function, while we use a convex $\ell_1$-norm penalty.

Although the algorithms we develop are valid for arbitrary filters $Q$, in this paper we only focus on gradient filters, *i.e*. the case when $Q$ is the (weighted) incidence matrix of $G$, such that rows of $Q\boldsymbol{y}$ give the (weighted) differences of neighboring labels. Thus:

$$E(\boldsymbol{y} \mid \mathcal{X}, \theta) = ||\boldsymbol{y} - \mathcal{X}\theta||_1 + \sum_{(i,j)\in\mathcal{E}} |w_{ij}\,(y_i - y_j)|, \tag{3}$$

where $w_{ij}$ are edge-weights. These edge-weights may themselves be functions of edge-features, *i.e*. $w_{ij} = \boldsymbol{x_{ij}}^T \beta$, where $\boldsymbol{x}_{ij}$ is a feature extracted at edge $(i, j)$, and $\beta$ is the (shared) edge parameter vector. For ease of explanation and to match our current implementation, in this paper we only describe learning techniques for $\theta$ and assume $w_{ij}$ to be known constants. However, the algorithm for learning $\beta$ is a straightforward generalization, and is presented in the supplementary material [1] (Section 1). Finally, note that we place no restrictions on the size or type of the graph-neighborhood and there could be arbitrary "long-range" links between the variables.

## 4. Learning and Inference in LCRFs

Before we go into details about parameter learning we need to describe inference in LCRFs.

### 4.1. Inference

We focus on maximum *a posteriori* inference in this model, which can be written as:

$$\hat{\boldsymbol{y}}(\theta) = \operatorname*{argmax}_{\boldsymbol{y}\in\mathbb{R}^n} P(\boldsymbol{y} \mid \mathcal{X}, \theta) \tag{4a}$$

$$= \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^n} \{||\boldsymbol{y} - \mathcal{X}\theta||_1 + ||Q\boldsymbol{y}||_1\} \tag{4b}$$

This is an $\ell_1$-norm minimization problem. It is well-known [4] that such problems may be formulated as a linear program. Let $\boldsymbol{q} \triangleq \mathcal{X}\theta$, $A \triangleq [\ I_{n\times n}\,;\, Q\ ]$ (where $I_{n\times n}$ is the $n \times n$ identity matrix) and $\boldsymbol{b} \triangleq [\ \boldsymbol{q}\,;\, \boldsymbol{0}_{m\times 1}\ ]$. Now:

$$\hat{\boldsymbol{y}}(\theta) = \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^n} \{||I\boldsymbol{y} - \boldsymbol{q}||_1 + ||Q\boldsymbol{y} - \boldsymbol{0}||_1\} \tag{5a}$$

$$= \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^n} ||A\boldsymbol{y} - \boldsymbol{b}||_1, \tag{5b}$$

$$= \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^n, \boldsymbol{\nu}\in\mathbb{R}^{(n+m)}} \boldsymbol{\nu}\cdot\boldsymbol{1} \tag{5c}$$

$$s.t. \quad \boldsymbol{\nu} \geq A\boldsymbol{y} - \boldsymbol{b}, \tag{5d}$$

$$\boldsymbol{\nu} \geq -(A\boldsymbol{y} - \boldsymbol{b}) \tag{5e}$$

where $\boldsymbol{\nu}$ is an auxiliary variable. Notice that (5c) is now a Linear Program (LP). The trick above is to notice that absolute value minimization can be replaced by two linear lower bounds. As we will see next, this trick helps us more than once.

### 4.2. Parameter Learning

Parameter learning involves finding the optimal values of parameter $\theta$ from labeled training data. Let us first consider a single training sample $(\mathcal{X}, \boldsymbol{y}^*)$, where $\boldsymbol{y}^*$ is the ground-truth labeling. We start with the margin-rescaled Structured SVM formulation of Tsochantaridis *et al*. [35], which minimizes the following problem:

$$\min_{\theta,\xi} \quad \frac{1}{2}||\theta||_2^2 + C\,\xi \tag{6a}$$

$$s.t. \quad E(\boldsymbol{y}^i|\mathcal{X}, \theta) - E(\boldsymbol{y}^*|\mathcal{X}, \theta) \geq \Delta(\boldsymbol{y}^i, \boldsymbol{y}^*) - \xi \qquad \forall \boldsymbol{y}^i \tag{6b}$$

$$\xi \geq 0, \tag{6c}$$

where $\Delta(\boldsymbol{y}^i, \boldsymbol{y}^*)$ is a user-specified risk function measuring the separation between labelings, and $C$ is a positive multiplier. Intuitively, we can see that SSVM minimizes a quadratic objective subject to constraints that enforce a soft margin between the energy of ground-truth and any other labeling, such that the margin is scaled by the risk function. It is known [35] that all of the above constraints (6b),(6c) may be expressed compactly via the *structured-hinge-loss*:

$$HLoss(\theta) = \max\Big\{0, E(\boldsymbol{y}^*|\mathcal{X}, \theta) - \min_{\boldsymbol{y}^i}\Big(E(\boldsymbol{y}^i|\mathcal{X}, \theta) - \Delta(\boldsymbol{y}^i, \boldsymbol{y}^*)\Big)\Big\}. \tag{7}$$

It can be shown [35] that hinge-loss is a convex upper-bound on risk incurred by the MAP solution $\hat{\boldsymbol{y}}$, *i.e*. $HLoss(\theta) \geq \Delta(\hat{\boldsymbol{y}}(\theta), \boldsymbol{y}^*)$. Thus, SSVMs can be understood as minimizing a regularized structured hinge loss:

$$\min_{\theta} \quad \frac{1}{2}||\theta||_2^2 + C\ HLoss(\theta) \tag{8}$$

Note that for a continuous-valued CRF, the set of all possible other labelings is an infinitely large set, and thus the above program cannot even be written down. Following the

work of Tsochantaridis *et al*. [35], we address this problem by using a cutting-plane approach. Specifically, we initialize this program with a small set of "bad" labelings $\tilde{\mathcal{I}}$, then learn $\theta$, and if the optimal labeling corresponding to this learnt $\theta$, *i.e*. the solution to (4), is not already in the set $\tilde{\mathcal{I}}$ (within some tolerance factor), we add it to the set and repeat. Formally, we repeatedly solve:

$$(MM:\tilde{\mathcal{I}}) \quad \min_{\theta,\xi} \frac{1}{2}||\theta||_2^2 + C\,\xi \tag{9a}$$

$$s.t. \quad ||\boldsymbol{y}^i - \mathcal{X}\theta||_1 + ||Q\boldsymbol{y}^i||_1 - ||\boldsymbol{y}^* - \mathcal{X}\theta||_1 - ||Q\boldsymbol{y}^*||_1 \geq 1-\xi, \quad \forall i \in \tilde{\mathcal{I}} \tag{9b}$$

$$\xi \geq 0. \tag{9c}$$

**Nonconvexity of Hinge-Loss.** Recall that in a typical SSVM or M$^3$N, the energy function is linear in parameters. In our case, the energy function contains $\ell_1$-norm terms, and thus constraints (9b) are not linear in $\theta$. Unfortunately, this makes the corresponding hinge-loss non-convex. Formally, we can state the following:

**Theorem 1** *Hinge Loss for the LCRF model,* i.e. $HLoss(\theta) = \max\Big\{0, ||\boldsymbol{y}^* - \mathcal{X}\theta||_1 + ||Q\boldsymbol{y}^*||_1 - \min_{\boldsymbol{y}^i}\Big(||\boldsymbol{y}^i - \mathcal{X}\theta||_1 + ||Q\boldsymbol{y}^i||_1 - \Delta(\boldsymbol{y}^i, \boldsymbol{y}^*)\Big)\Big\}$, *is non-convex in* $\theta$.

**Proof.** See Supplementary Material [1], Section 2.

Due to this non-convexity, standard techniques like sub-gradient descent and cutting-plane methods cannot be used to minimize the LCRF hinge-loss. Furthermore, note that the variables $\theta$ multiply with $\mathcal{X}$ and therefore every absolute value term contains all the components of $\theta$. This does not allows use of search algorithms such as in [15].

**Approximate Max-Margin Learning.** We now show how the non-convex program $(MM:\tilde{\mathcal{I}})$ can be *approximated* by a convex QP. The following exposition is described with a 0-1 risk, however any risk-function (*e.g*. hamming) may be used as long as the *risk-augmented energy minimization problem* [35] is tractable. We use the same trick as we did in (5c) to convert an $\ell_1$-norm minimization into an LP, using auxiliary variables: $\boldsymbol{d}^*, \{\boldsymbol{d}^i\} \in \mathbb{R}^n$:

$$(MMQP:\tilde{\mathcal{I}})$$

$$\min_{\theta,\xi,\{\boldsymbol{d}^*\},\{\boldsymbol{d}^i\}} \quad \frac{1}{2}||\theta||_2^2 + C\,\xi + C_1\sum_{j=1}^{n} d_j^* + C_2\sum_{i\in\tilde{\mathcal{I}}}\sum_{j=1}^{n} d_j^i \tag{10a}$$

$$s.t. \quad \sum_{j=1}^{n} d_j^i - \sum_{j=1}^{n} d_j^* \geq 1 + ||Q\boldsymbol{y}^*||_1 - ||Q\boldsymbol{y}^i||_1 - \xi \tag{10b}$$

$$\xi \geq 0 \qquad \forall i \in \tilde{\mathcal{I}} \tag{10c}$$

$$\boldsymbol{d}^* \geq +(\boldsymbol{y}^* - \mathcal{X}\theta), \quad \boldsymbol{d}^* \geq -(\boldsymbol{y}^* - \mathcal{X}\theta) \tag{10d}$$

$$\boldsymbol{d}^i \geq +(\boldsymbol{y}^i - \mathcal{X}\theta), \quad \boldsymbol{d}^i \geq -(\boldsymbol{y}^i - \mathcal{X}\theta) \quad \forall i \in \tilde{\mathcal{I}} \tag{10e}$$

where $C_1, C_2$ are positive weights (see [1] for how to set them). All constraints in the above program $(MMQP:\tilde{\mathcal{I}})$ are linear in $\theta, \xi, \{\boldsymbol{d}^*\}, \{\boldsymbol{d}^i\}$, and this program is a convex quadratic program, solvable by standard techniques. Formally, we can state the following about this approximation:

**Theorem 2** *If* $\{\hat{\theta}, \hat{\xi}, \hat{\boldsymbol{d}}^*, \hat{\boldsymbol{d}}^i\}$ *is the optimum solution of* $MMQP:\tilde{\mathcal{I}}$ *(10), then* $\hat{\xi}$ *is equal to the LCRF hinge-loss* $HLoss(\hat{\theta})$*, and thus an upper-bound on the loss incurred by the MAP solution,* i.e. $\hat{\xi} = HLoss(\hat{\theta}) \geq \Delta(\hat{\boldsymbol{y}}(\hat{\theta}), \boldsymbol{y}^*)$.

**Proof.** See Supplementary Material [1], Section 3.

Thus, the constraints of the two programs – $MM:\tilde{\mathcal{I}}$ (9) and $MMQP:\tilde{\mathcal{I}}$ (10) – represent *exactly* the same object, *i.e*. structured hinge-loss. The approximation comes from the extra terms in the objective function (10a), which are necessary for linearizing the $\ell_1$-norm terms.

From a computational perspective, it is important to point out one drawback of this linearization approach. Program $(MMQP:\tilde{\mathcal{I}})$ includes vector constraints (10d),(10e) of dimension equal to the number of random variables ($n$). While constraint (10d) does not grow with iterations of the cutting-plane algorithm, constraint (10e) does. Thus, each additional "bad" labeling added to the list $\tilde{\mathcal{I}}$ adds $n$ more constraints to the QP, which may become impractical. However, as we see next, we use ideas from the dual-decomposition [3, 7] literature to restrict this QP to a manageable size.

### 4.3. Extension to Multiple Training Images via Dual-Decomposition

Let us now extend our algorithm to learn from multiple images. Consider a training dataset indexed by $\mathcal{T} = \{1, 2, \ldots, T\}$. Let $\xi^{(t)}$ denote the slack variable and vector $\boldsymbol{D}^{(t)} = \{\boldsymbol{d}^{*(t)}, \boldsymbol{d}^{i\,(t)} \mid i \in \tilde{\mathcal{I}}\}$ hold all auxiliary variables for a training image $t$. For brevity of description, let us denote all linear constraints in (10) with the polytope $\mathcal{P}^{(t)}$. Thus, $\{\theta, \xi^{(t)}, \boldsymbol{D}^{(t)}\} \in \mathcal{P}^{(t)}$ denotes the set of solutions feasible according to image $t$. We can now write down a straightforward generalization of the previous program $(MMQP:\tilde{\mathcal{I}})$ to multiple training images:

$$(MMQP:\tilde{\mathcal{I}}^{\mathcal{T}})$$

$$\min_{\theta,\{\xi^{(t)},\boldsymbol{D}^{(t)}\}} \quad \frac{1}{2}||\theta||_2^2 + \frac{C}{T}\sum_{t\in\mathcal{T}}\xi^{(t)} + \frac{C'}{T}\sum_{t\in\mathcal{T}}\boldsymbol{D}^{(t)}\cdot\mathbf{1} \tag{11a}$$

$$s.t. \qquad \{\theta, \xi^{(t)}, \boldsymbol{D}^{(t)}\} \in \mathcal{P}^{(t)} \qquad \forall t \in \mathcal{T}. \tag{11b}$$

Clearly, as the size of the training dataset increases, this program becomes larger, and very quickly impractical. The key here is to notice that this large program $(MMQP:\tilde{\mathcal{I}}^{\mathcal{T}})$ consists of several *almost* independent problems over individual images, only coupled by the parameter $\theta$. We follow a dual-decomposition approach, where we solve a Lagrangian relaxation of this problem which easily decomposes to smaller independent sub-problems for each training image. This enables solving the problem over a distributed architecture or a cloud of machines, and thus scales well to large datasets. Most importantly, we prove this Lagrangian relaxation achieves zero duality gap and in fact is a tight relaxation (see Theorem 3). Thus, the solution to the Lagrangian relaxation converges to the solution of $MMQP:\tilde{\mathcal{I}}^{\mathcal{T}}$.

We describe this relaxation next. First, we reparameterize the previous program by allocating to each training image its own copy of the parameters $\theta^{(t)}$:

$$(MMQP : \tilde{\mathcal{I}}^{\mathcal{T}}2)$$

$$\min_{\tilde{\theta},\{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}\}} \quad \frac{1}{2T}\sum_{t\in\mathcal{T}}||\theta^{(t)}||_2^2 + \frac{C}{T}\sum_{t\in\mathcal{T}}\xi^{(t)} + \frac{C'}{T}\sum_{t\in\mathcal{T}}\boldsymbol{D^{(t)}}\cdot\mathbf{1} \tag{12a}$$

$$s.t. \quad \{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}\} \in \mathcal{P}^{(t)} \tag{12b}$$

$$\theta^{(t)} = \tilde{\theta} \qquad \forall t \in \mathcal{T}. \tag{12c}$$

The above program $(MMQP : \tilde{\mathcal{I}}^{\mathcal{T}}2)$ uses a global variable $\tilde{\theta}$ to force all training images to have the same parameters, and thus is equivalent to the earlier program $(MMQP : \tilde{\mathcal{I}}^{\mathcal{T}})$. However, we can now relax constraints (12c). We consider a Lagrangian relaxation of the above program by *dualizing* [7] the complicating constraints (12c):

$$(LR : \tilde{\mathcal{I}}^{\mathcal{T}})$$

$$\min_{\tilde{\theta},\{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}\}} \quad \frac{1}{2T}\sum_{t\in\mathcal{T}}||\theta^{(t)}||_2^2 + \frac{C}{T}\sum_{t\in\mathcal{T}}\xi^{(t)} + \frac{C'}{T}\sum_{t\in\mathcal{T}}\boldsymbol{D^{(t)}}\cdot\mathbf{1} + \sum_{t\in\mathcal{T}}\boldsymbol{\lambda}^{(t)}\cdot(\theta^{(t)}-\tilde{\theta}) \tag{13a}$$

$$s.t. \quad \{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}\} \in \mathcal{P}^{(t)} \qquad \forall t \in \mathcal{T}, \tag{13b}$$

where $\boldsymbol{\lambda}^{(t)} \in \mathbb{R}^k$ are the (unconstrained) Lagrangian multipliers, which may be thought of as indicating the penalties or costs for violating their corresponding constraints. Note that now the above problem is completely separable into independent sub-problems for each training image.

$$(LR : \tilde{\mathcal{I}}^{\mathcal{T}}2)$$

$$\sum_{t\in\mathcal{T}}\min_{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}} \Big(\frac{1}{2T}||\theta^{(t)}||_2^2 + \boldsymbol{\lambda}^{(t)}\cdot\theta^{(t)} + \frac{C}{T}\xi^{(t)} + \frac{C'}{T}\boldsymbol{D^{(t)}}\cdot\mathbf{1}\Big) - \min_{\tilde{\theta}}\Big(\sum_{t\in\mathcal{T}}\boldsymbol{\lambda}^{(t)}\Big)\cdot\tilde{\theta} \tag{14a}$$

$$s.t. \quad \{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}\} \in \mathcal{P}^{(t)} \qquad \forall t \in \mathcal{T}. \tag{14b}$$

We can see that the (unconstrained) optimization over $\tilde{\theta}$ forces a constraint on the Lagrangian variables, *i.e.* $\sum_{t\in\mathcal{T}}\boldsymbol{\lambda}^{(t)} = 0$; otherwise the program will not have a finite value. Let us now define these independent sub-problems as a function of the dual variables $(\boldsymbol{\lambda}^{(t)})$:

$$\mathcal{F}^{(t)}(\boldsymbol{\lambda}^{(t)}) = \min_{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}} \frac{1}{2T}||\theta^{(t)}||_2^2 + \boldsymbol{\lambda}^{(t)}\cdot\theta^{(t)} + \frac{C}{T}\xi^{(t)} + \frac{C'}{T}\boldsymbol{D^{(t)}}\cdot\mathbf{1} \tag{15a}$$

$$s.t. \quad \{\theta^{(t)},\xi^{(t)},\boldsymbol{D^{(t)}}\} \in \mathcal{P}^{(t)}. \tag{15b}$$

We can now search for the tightest relaxation by optimizing over the dual variables $\{\boldsymbol{\lambda}^{(t)}\}$. Formally, this is the Lagrangian dual of $(MMQP : \tilde{\mathcal{I}}^{\mathcal{T}})$:

$$(LD : \tilde{\mathcal{I}}^{\mathcal{T}}) \quad \max_{\{\boldsymbol{\lambda}^{(t)}\}} \sum_{t\in\mathcal{T}}\mathcal{F}^{(t)}(\boldsymbol{\lambda}^{(t)}) \tag{16a}$$

$$s.t. \quad \sum_{t\in\mathcal{T}}\boldsymbol{\lambda}^{(t)} = 0. \tag{16b}$$

**Algorithm 1.** We solve this dual problem via projected gradient ascent. It is easy to verify that the gradient of each sub-problem with respect to the Lagrangian multiplier is simply the optimal parameter learned from that sub-problem, *i.e.* $\frac{\partial\mathcal{F}^{(t)}}{\partial\boldsymbol{\lambda}^{(t)}} = \hat{\theta}^{(t)}$, where $\hat{\theta}^{(t)}$ is the optimal solution to problem (15). We also note that each subproblem is strongly convex in $\theta^{(t)}$ and thus has a unique optimum $\hat{\theta}^{(t)}$. The projection step is fairly simple – it involves satisfying the zero-mean constraint of (16b), which can be enforced by subtracting the mean of the dual variables. Overall, the update rule is given by $\boldsymbol{\lambda}^{(t)} \longleftarrow \Big[\boldsymbol{\lambda}^{(t)} + \alpha\frac{\partial\mathcal{F}^{(t)}}{\partial\boldsymbol{\lambda}^{(t)}}\Big]_0$, where $\alpha$ is the step-size and $[\cdot]_0$ is the projection operator, *i.e.* $\big[x^{(t)}\big]_0 = x^{(t)} - \frac{1}{T}\sum_{t\in\mathcal{T}}x^{(t)}$. We can see that each step of gradient ascent requires learning parameters on all sub-problems. Thus, we have converted a large QP into several smaller QPs that can be independently optimized in parallel, although they need to be solved several times.

Most importantly, the following theorem shows that the Lagrangian relaxation does *not* introduce a second level of approximation In fact, our algorithm *exactly* solves $MMQP : \tilde{\mathcal{I}}^{\mathcal{T}}$.

**Theorem 3** *$LD : \tilde{\mathcal{I}}^{\mathcal{T}}$ (16) has zero duality gap and Algorithm 1 converges to the optimum of $MMQP : \tilde{\mathcal{I}}^{\mathcal{T}}$ (11).*
**Proof.** See Supplementary Material [1], Section 4.

## 5. Experiments

We apply our learning algorithm to the problem of estimating depth from a single image (see Fig. 2 for examples). This is a difficult mathematically-ill-posed problem due to the ambiguities introduced by the projection of the 3D world onto a 2D image. Local features alone are typically not enough for estimating depth (e.g., both sky and water can be blue, a gray patch could be sidewalk or a wall). We need to use a CRF to model the relations between the depths at neighboring regions. Hand-tuning weights on features is difficult and impractical and thus a good learning algorithm is essential for this application.

For an $h \times w$ image, our model variables $\{y_i \mid i \in [n], n = hw\}$ indicate log-depth at each pixel, given image features $\boldsymbol{x}_i \in \mathbb{R}^k$ at the corresponding pixels in the image. We follow the same parameter sharing scheme as Saxena *et al.* [27], *i.e.* assign each row of an image its own parameter vector. We use two kinds of features: 105-dim texture features computed using the publicly-available code from Make3D [24], and 8-dim semantic-category-prediction features at each pixel provided by Liu *et al.* [20]. When we compare to other works using the same features, the performance difference can be attributed to the choice of model and learning algorithm used. Moreover, Saxena *et al.* [24] used the same (texture) features and model as us (LCRF), and trained the parameters simply by minimizing the $\ell_1$ error $||y - \mathcal{X}\theta||_1$. Any performance difference between their

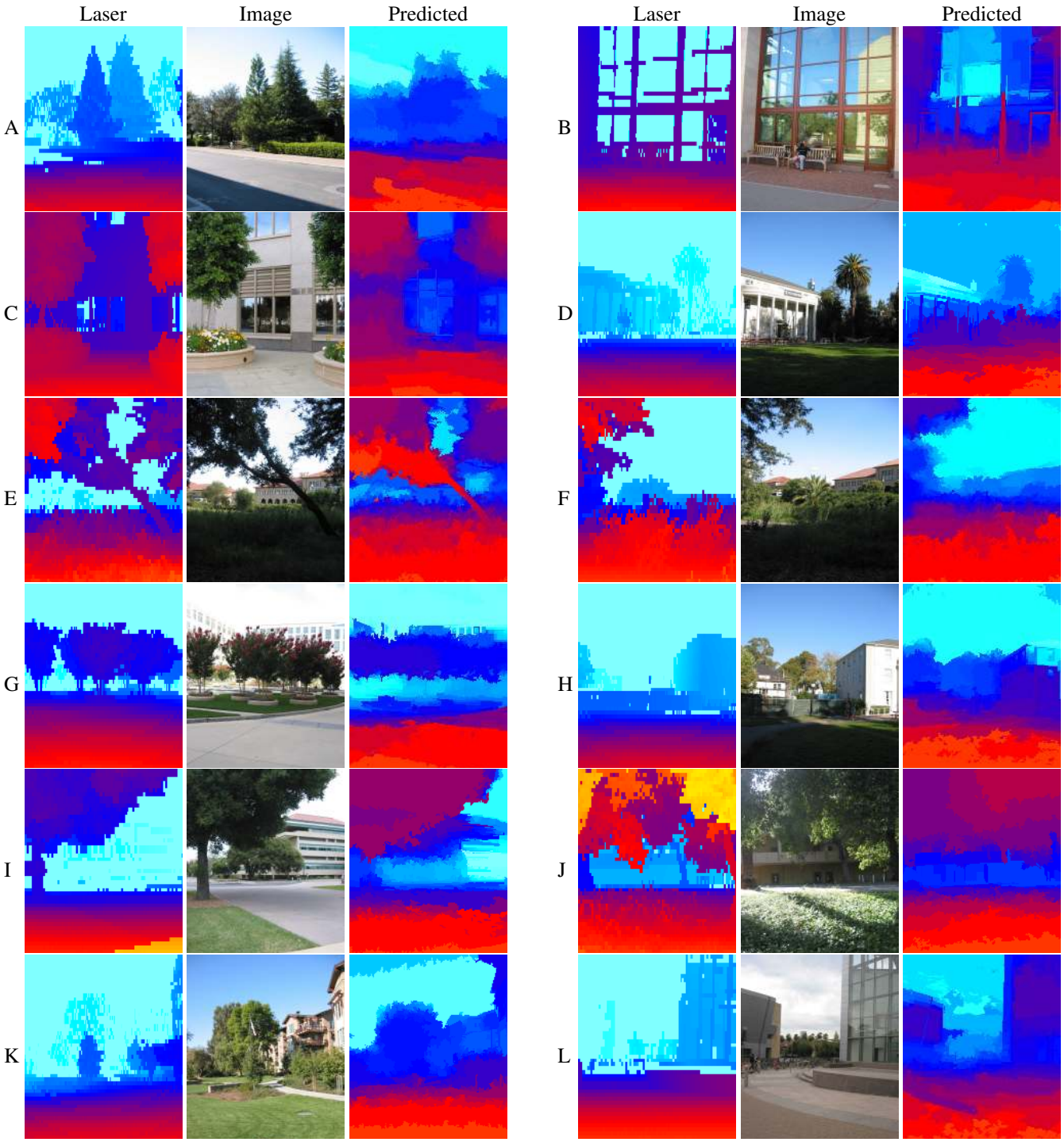
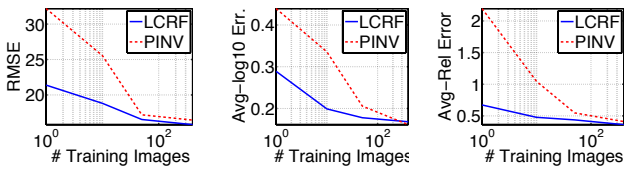



Figure 2: Results on single-image depth estimation using LCRFs trained with our learning algorithm. (Left) Laser ground-truth depths. (Middle) Original image. (Right) Predicted depths. We can see that LCRFs model depth discontinuities well. **(Best viewed in color.)**

work and ours is specifically attributed to the use of our max-margin learning algorithm.

We test our approach on the Make3D Range Image dataset [25, 27], which consists of 534 images (400 training, 134 testing) with ground-truth depths obtained from a laser scanner. The variety of environments (roads, buildings, trees, indoor corridors, *etc*.) presents situations such as sharp depth changes (due to occlusions) and thin long struc-




(a) RMSE (b) Avg-log10 Error (c) Avg-Rel Error

Figure 3: Error on test set vs the number of training images. (Red) Pseudo-Inverse training. (Blue) Our training algorithm. Note that our training method performs well even with a small number of training images.

tures (trees, poles, *etc*.), making this a challenging dataset. All results are reported on the 134 test images.

We measure our performance with a commonly used error metric on this dataset called `rel-error`, defined as $\frac{1}{n}\sum_{j=1}^{n}(\hat{y}_j - y_j^*)/y_j^*$, where $y_j^*$ is the ground-truth depth for pixel $j$, and $\hat{y}_j$ is the predicted depth. Another error metric that is also sometimes used is `log10` error metric $\frac{1}{n}\sum_{j=1}^{n}|\log_{10}\hat{y}_j - \log_{10} y_j^*|$. For the sake of completeness, we also report `RMSE` errors $\sqrt{\frac{1}{n}\sum_{j=1}^{n}(\hat{y}_j - y_j^*)^2}$, which a number of previous works do not report.

We experimented with decomposition into sub-problems of various sizes, and found decomposition into 8 fifty-image sub-problems to be an ideal choice. Note that the zero duality gap statements in Theorem 3 hold for batch decomposition as well. We typically ran Algorithm 1 for 4-8 steps, and typical cardinality of set $\tilde{\mathcal{I}}$ was 5-10.

**Comparison to State-of-the-Art.** We compare our results to a number of other works that have reported results on this dataset. The comparison is shown in Table 1. The results for "Chance" baseline are taken from Saxena *et al*. [27]. This table also lists the major improvement source over the original work of Saxena *et al*. [24] (SCN). We note that a direct comparison with these methods is problematic. Some methods use more sophisticated models and sometimes additional data/features, limiting the conclusions that may be drawn from the comparison. For example SCN [24] use a hierarchical MRF; Liu *et al*. [20] (LGK) use semantic labels and additional geometry based priors in the CRF. Heitz *et al*. [9] and Li *et al*.'s [16] cascaded classification models combine information from object detection, image segmentation and scene categorization. Overall, these works focused on using context in order to improve performance. On the other hand, our method only uses a single 4-connected grid and no additional information. Despite the simplicity, our model achieves 0.362 rel-error, which is better than current best of 0.370. We attribute this superior performance to our learning algorithm, which enables training an accurate model (LCRF) for the problem. On RMSE and log10 errors, our approach is competitive with the state of the art, but not the lowest. Hopefully, performance can be further improved by combining these orthogonal ideas – our LCRF model and learning algorithm, semantic modeling of LGK, and multi-task information of Heitz *et al*.

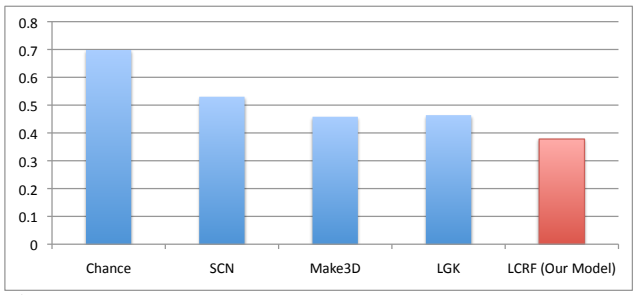



Figure 4: Effects of learning algorithms on `pointwise-CRF` with `Make3D features`. Plotting Avg-Rel Error. This figure shows that even with simple features, our learning technique improves the performance over other methods.

**Stability w.r.t. Small Training Set.** We also analyzed the performance of our learning algorithm with the size of training set. Fig. 3 shows that our algorithm is less prone to overfitting as compared to the pseudo-inverse training of SCN, *i.e*. even with small amounts of training data, it can perform reasonably well on the test set.

**Effect of Learning Algorithm.** Finally, we try to isolate the the effect of the learning algorithm from other factors. To do this, we only compare baselines that all use exactly the same features and approximate the same MRF/CRF structure (although different kinds of potentials). These include SCN [24], the "pointwise CRF" from Make3D [27] and the "Pixel CRF model" from LGK [20]. For this comparison, we trained our model with texture features only. Fig. 4 shows the results. We see that our learning algorithm performs significantly better than SCN, LGK and Make3D.

**Qualitative Results.** In Fig. 2, we show the predicted depths for a few examples in the test set, together with the laser ground-truth depths for comparison. Our algorithm makes quite reasonable-looking predictions. In general, CRFs suffer from the problem of over-smoothing (e.g. observed by Saxena *et al*. [24, 27]). However, this problem seems to be less acute in our method—we believe this is because our learning method learns the parameters while taking into account the edge terms in the CRF, and thus results in sharper (see Fig. 2-E,H,I) and more accurate depths. The problem still persists in some cases, such as in Fig. 2-J, where our algorithm was confused by the texture of the leaves, and produced over-smoothed depths.

Note that the ground-truth labels were limited to a range of 80 meters, and therefore, in most of the images in Fig. 2, we see that far-away structures are measured as 80m (same as sky) by the laser. Our model reasonably predicts even far-away parts in the image (and the actual ground-truth label is wrong!). See Fig. 2-A,D,F,G,H,I. In Fig. 2-B, we see the reflections of another building, trees and sky into a glass-paned transparent wall. The laser scanner measures depths incorrectly because the pulses get scattered by the

Table 1: Summary of results for the depth estimation task. Empty entrees indicate that those numbers were not reported in the prior work. Note that different methods use different features, different structure of the MRFs/CRFs, as well as other additional information in some cases. See Fig. 4 for the effect of learning algorithm alone. See text for details.

| Method | Description (main improvement source) | RMSE-linear | Avg-log10 | Avg-Rel |
|---|---|---|---|---|
| Chance | predict mean depthmap | 28 | 0.334 | 0.698 |
| SCN [24] | hierarchical, pointwise CRF, Laplacian potentials | 16.7 | 0.198 | 0.530 |
| HEH [11] | surface layout, discrete CRF | - | 0.320 | 1.423 |
| Make3D - pointwise CRF [27] | tertiary connections | - | 0.149 | 0.458 |
| Make3D - superpixel CRF [26] | superpixel formulation | - | 0.187 | 0.370 |
| LGK - pointwise CRF [20] | semantic segmentation, geometry | - | 0.149 | 0.375 |
| LGK - superpixel CRF [20] | semantic segmentation, geometry | - | **0.148** | 0.379 |
| CCM [9] - cascaded models | object detection, segmentation, categorization | 15.4m | - | - |
| Li *et al*. [16] - feedback cascades | object detection, categorization, event, geometric layout | **15.2m** | - | - |
| Li *et al*. [17] | probabilistic dependence between parameters | **15.2m** | - | - |
| Our model - pointwise CRF | max-margin learning | 15.8 | 0.168 | **0.362** |

glass. Our algorithm relies on the image and estimates the depth of the reflected structures instead.

## 6. Conclusions

In this paper, we considered continuous-valued CRFs with heavy-tailed Laplacian potentials. Although LCRFs are the ideal modeling choice for many applications and inference in these models is convex and tractable, parameter learning could only be performed heuristically in prior work. We presented the first (approximate) max-margin parameter learning algorithm for LCRFs, by linearizing the non-convex $\ell_1$-norm constraints. We also presented a dual-decomposition-based algorithm to make learning scalable in the number of training images.

Future work involves exploring more sophisticated decomposition techniques like Augmented Lagrangian methods. In addition, $\ell_1$-norm minimization problems are often convex relaxations for solving $\ell_0$-norm minimization *i.e.* cardinality constraints. We believe that the learning algorithm for LCRFs presented in this paper could be useful for MRFs/CRFs with such potentials as well.

## References


[1] Authors. Please see supplementary materials. 3, 4, 5

[2] A. Barbu. Learning real-time mrf inference for image denoising. In *CVPR*, 2009. 1, 2

[3] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999. 2, 4

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. 3

[5] B. T. Carlos, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 2003. 2

[6] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 2

[7] M. Guignard. Lagrangean relaxation. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 11(2):151–200, 2003. 2, 4, 5

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2009. 2

[9] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, 2008. 7, 8

[10] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002. 2

[11] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 8

[12] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. 1

[13] J. Huang, A. Lee, and D. Mumford. Statistics of range images. In *CVPR*, 2000. 1

[14] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009. 1

[15] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006. 4

[16] C. Li, A. Kowdle, A. Saxena, and T. Chen. Towards holistic scene understanding: Feedback enabled cascaded classification models. In *NIPS*, 2010. 2, 7, 8

[17] C. Li, A. Saxena, and T. Chen. $\theta$-mrf: Capturing spatial and semantic structure in the parameters for scene understanding. In *NIPS*, 2011. 8

[18] Y. Li and D. P. Huttenlocher. Learning for optical flow using stochastic optimization. In *ECCV*, 2008. 2

[19] Y. Li and D. P. Huttenlocher. Learning for stereo vision using the structured support vector machine. *CVPR*, 2008. 2

[20] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, 2010. 2, 5, 7, 8

[21] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996. 1

[22] S. Roth and M. Black. Fields of experts. *IJCV*, 82(2), April 2009. 1, 2

[23] K. Samuel and M. Tappen. Learning optimized MAP estimates in continuously-valued MRF models. In *CVPR*, 2009. 1, 2

[24] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *In NIPS 18*, 2005. 1, 2, 5, 7, 8

[25] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 2007. 1, 6

[26] A. Saxena, M. Sun, and A. Y. Ng. Learning 3-d scene structure from a single still image. In *ICCV workshop on 3D Representation for Recognition (3dRR-07)*, 2007. 8

[27] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3D scene structure from a single still image. *PAMI*, 31:824–840, 2009. 2, 5, 6, 7, 8

[28] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR*, 2007. 2

[29] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on mrfs in low-level vision. In *CVPR*, 2010. 1

[30] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *CVPR*, 2006. 2

[31] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *ECCV*, 2008. 2

[32] M. F. Tappen. Utilizing variational optimization to learn markov random fields. In *CVPR*, 2007. 1, 2, 3

[33] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning gaussian conditional random fields for low-level vision. In *CVPR*, 2007. 2

[34] A. Torralba and A. Oliva. Statistics of natural image categories. *Comput. Neural Syst.*, 14:391–412, 2003. 1

[35] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005. 1, 2, 3, 4

[36] Y. Weiss and W. T. Freeman. What makes a good model of natural images? *CVPR*, 2007. 1, 2

[37] J. Zhu, E. P. Xing, and B. Zhang. Laplace maximum margin markov networks. In *ICML*, 2008. 3

[38] S. C. Zhu and D. Mumford. Prior learning and gibbs reaction-diffusion. *PAMI*, 19(11):1236–1250, 1997. 1