Structured Regression Gradient Boosting

Ferran Diego Fred A. Hamprecht
Heidelberg Collaboratory for Image Processing (HCI)
Interdisciplinary Center for Scientific Computing (IWR)
University of Heidelberg, Heidelberg 69115, Germany
{ferran.diego,fred.hamprecht}@iwr.uni-heidelberg.de

Abstract

We propose a new way to train a structured output prediction model. More specifically, we train nonlinear data terms in a Gaussian Conditional Random Field (GCRF) by a generalized version of gradient boosting. The approach is evaluated on three challenging regression benchmarks: vessel detection, single image depth estimation and image inpainting. These experiments suggest that the proposed boosting framework matches or exceeds the state-of-the-art.

1. Introduction

Many problems in machine learning involve the prediction of outputs that are not a single value, but a more complicated object like a sequence, a graph or an image. Such problems are referred to as *structured output prediction* [26]. Markov random fields have become popular for structured prediction thanks to their ability to exhibit desirable global behavior based on the specification of local or sparse interactions only. For generative models, the parameters can be found by maximum likelihood [46, 19, 6, 4]; and for both generative and conditional models by discriminative training [43, 44, 12, 15]. The latter is generally found to work better except when training data is extremely scarce. This success is at least partially attributed to the fact that such models learn to minimize the loss function of interest for a specific task, rather than solve a more general problem, namely learning to generate images. The most frequent approach to discriminative learning of structured models is via max margin (structSVM, [47, 45]). One limitation of structSVMs is that the model must be linear in its parameters in a joint feature space. Recent alternatives include regression tree fields [12, 14], where nonparametric potentials are learned by minimizing a loss function using a projected gradient method.

In this paper we propose, for the first time, a generalization of gradient boosting that allows directly training a structured regression model and exploit all its benefits for structured regression tasks in a principled way (Sect. 2). The procedure, summarized in Fig. 1 and Algorithm 1, is described in detail in section 2. In section 3, we study performance on a number of benchmarks for which many papers have previously pushed the accuracy to levels that are remarkable given the difficulty of those problems. We close with a brief summary in section 4.

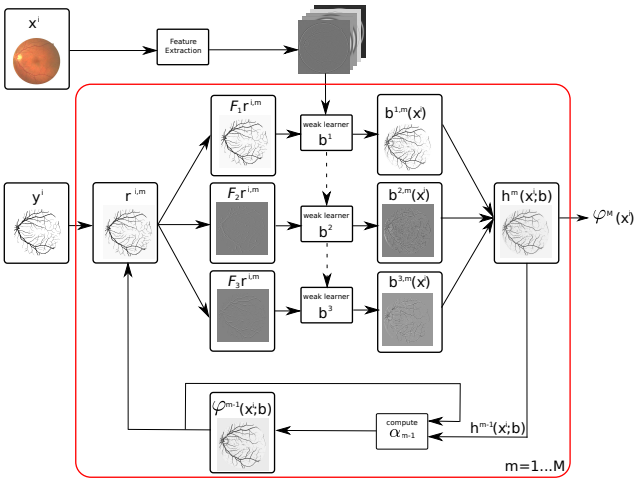

Figure 1. Structured regression gradient boosting. Given input x^i and structured ground truth y^i, we iterate the following: given the current prediction $\varphi^{m-1}(\mathrm{x}^i)$, compute gradient $\mathrm{r}^{i,m}$ of the structured loss. Next, train shallow regression trees that produce good fits $\mathrm{b}^{k,m}$ to the structured loss gradient image and filtered versions thereof. Use these fits to parametrize the data terms of a Gaussian conditional random field. Its MAP solution $\mathrm{h}^m(\mathrm{x}^i)$ is added to the current strong structured learner with a weight α_m found through line search. Repeat M times.

1.1. Related Work

Discrete and continuous conditional random fields (CRFs) [18, 33, 43, 12] are structured prediction models that model explicitly the relation among latent variables. Learning the optimal parameters of these models has been

studied extensively in the literature. StructSVMs [47, 45] have been applied to learn the parameters of a linear joint feature function over the input-output pairs while maximizing a margin [42]. In contrast, non-linear mappings on joint feature functions have been learnt using gradient boosting while minimizing the negative conditional log-likelihood [46, 19, 6, 4]. Following the idea of discriminative learning, few methods [30, 29, 37] have been proposed for structured regression that share only partial ideas with the Gradient Boosting framework. Ratliff *et al*. [30] add a new feature as a nonlinear function of the original base features using a weak classifier trained from a previous prediction. Parker *et al*. [29] instead reformulates the structured perceptron to reweight the training set at each iteration as in AdaBoost. StructBoost [37] incorporates only the notion of weak structured predictors into structSVM that are generated by finding the most violated constraint. In contrast, the proposed method is the unique method that extends gradient boosting framework to structured output prediction as an ensemble of weak structured learners while keeping the original formulation as a specific case.

Closest in spirit to our work is StructBoost [37] which supports a nonlinear structured learning by combining a set of weak non-linear structured learners. This approach differs from the proposed approach in the following points. First, StructBoost generalizes AdaBoost or LPBoost to structured learning; our approach is based on gradient boosting which is in itself a generalization of Adaboost. Second, the weak structured learner of StructBoost is a function that maps an input-output pair to a scalar value that measures the compatibility of the input and output. Instead, our proposed weak structured learner maps the observed variable to a structured output, not just a compatibility measure. Third, StructBoost is formulated to maximize a margin; in contrast, our method minimizes an empirical risk without maximizing the margin as in GCRFs [12, 43] that are the state–of–the–art on some tasks such as image denoising. Finally, our proposed weak learner extends GCRF [33] to be more expressive by generalizing the data term as a set of convolutional kernels and by learning a non–parametric regression tree used in [12].

The term "Structured Gradient Boosting" has been used before in [29], though with a completely different goal: learning the parameters of structured perceptron algorithm [5] that is reformulated to reweight the training set at each iteration as in AdaBoost; there, the structure prediction is relegated to the structured perceptron without learning any weak learner.

2. Method

We build on the gradient boosting framework, but unlike the original formulation use *structured* output weak learners. Gradient boosting aims at approximating a function $\phi^* : \mathbb{R}^p \rightarrow \mathbb{R}$ by a linear combination of weak learners. Instead, we propose Structured Regression Gradient Boosting (SRGB) to approximate a mapping $\varphi^* : \mathbb{R}^p \rightarrow \mathbb{R}^q$ by a function φ^M of the form

$$\varphi^M(\mathrm{x}) = \sum_{t=1}^{M} \alpha_t \mathrm{h}^t(\mathrm{x}). \tag{1}$$

Here, $\alpha_t \in \mathbb{R}$ are real-valued weights, $\mathrm{h}^t : \mathbb{R}^p \rightarrow \mathbb{R}^q$ are weak structured regression predictors and $\mathrm{x} \in \mathbb{R}^p$ is the input vector, *e.g*. an observed image or all features derived from an image.

Given structured training exemplars $\{(\mathrm{x}^i, \mathrm{y}^i)\}_{i=1}^N$, where $\mathrm{x}^i \in \mathbb{R}^p$ and $\mathrm{y}^i \in \mathbb{R}^q$, we aim to minimize an empirical risk function

$$\mathcal{L}\left(\varphi^M\right) = \sum_{i=1}^{N} L\left(\mathrm{y}^i, \varphi^M(\mathrm{x}^i)\right). \tag{2}$$

Here, $L\left(\mathrm{y}, \varphi(\mathrm{x})\right)$ is a loss function that is differentiable w.r.t. each dimension of the predicted structured output. The risk function $\mathcal{L}$ in (2) is minimized in a greedy manner by iteratively adding up weak structured regression predictors as in the Gradient Boosting framework [9]. Other approaches such as structSVM endow the r.h.s. of Eq. (2) with an additional regularization term. In our case, regularization is accomplished by restrictions on the set of admissible weak learners, and by choosing a suitable number of iterations M.

Boosting updates can be interpreted as first-order approximations of the gradient descent direction in function space [9]. At each iteration m, we seek for the weak structured learner h^m that best predicts the average (negative) gradient direction to minimize the empirical risk function $\mathcal{L}(\varphi^m)$.

Denote by

$$\mathrm{r}^{i,m} = -\frac{\partial L(\mathrm{y}^i, \varphi^{m-1}(\mathrm{x}^i))}{\partial \varphi^{m-1}(\mathrm{x}^i)} \tag{3}$$

the column vector summarizing the negative gradient direction of the loss given the current prediction w.r.t. all dimensions of the current structured output prediction. We then want to solve, in each boosting iteration,

$$\mathrm{h}^m = \underset{\mathrm{h}(\cdot)\in\mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2} \left|\left|\mathrm{h}(\mathrm{x}^i) - \mathrm{r}^{i,m}\right|\right|_F^2 \tag{4}$$

where $||\cdot||_F$ is the Frobenius norm. The weak learners $\mathrm{h} \in \mathcal{H}$ in our case correspond to Gaussian Conditional Random Fields with potentials encoded by nonparametric regression trees (see Section 2.2). A nice feature of this choice is that the final predictor (the strong learner) is just a sum of GCRFs and is thus itself a single GCRF, affording very fast inference at test time.

An important contribution is the way in which the parameters of this GCRF are learned discriminatively, namely by simplifying the minimization of Eq. (4) in terms of an alternative quadratic formulation that has the same global minimum as the original formulation but which can be found in closed form, without resorting to projected gradient descent.

Once a weak learner h_m has been found, its weight α_m in the final predictor is found through line search:

$$\alpha_m = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(\mathrm{y}^i, \varphi^{m-1}(\cdot) + \alpha \mathrm{h}^m(\mathrm{x}^i)\right). \quad (5)$$

For the loss function, we adopt standard choices from Gradient Boosting, namely the exponential loss $L\left(\mathrm{y}^i, \varphi(\mathrm{x}^i)\right) = \sum_{j=1}^{q} e^{-\mathrm{y}_j^i \varphi_j(\mathrm{x}^i)}$ and the log loss $L\left(\mathrm{y}^i, \varphi(\mathrm{x}^i)\right) = \sum_{j=1}^{q} \log(1 + e^{-2\mathrm{y}_j^i \varphi_j(\mathrm{x}^i)}$ for binary classification, or the l_2-norm $L\left(\mathrm{y}^i, \varphi(\mathrm{x}^i)\right) = \sum_{j=1}^{q} (\mathrm{y}_j^i - \varphi_j(\mathrm{x}^i))^2$ for regression. In these definitions, $\varphi_j(\mathrm{x}^i)$ is the j^{th} element of the structured prediction function $\varphi(\mathrm{x}^i)$, and y_j^i is the corresponding element of the structured training label y^i.

2.1. Weak structured regressor: GCRF

Denote with $\mathrm{x} \in \mathbb{R}^p$ an observed image, and with $\mathrm{z} \in \mathbb{R}^p$ its corresponding labelling.

We model the conditional probability with a Gaussian random field with parameters $\mathcal{W}$,

$$p(\mathrm{z}|\mathrm{x}; \mathcal{W}) \propto \exp\left(-\frac{1}{2}(\mathrm{z} - \mu(\mathrm{x}))^T P (\mathrm{z} - \mu(\mathrm{x}))\right). \quad (6)$$

Note that in our experiments, we choose to let the precision matrix P be a constant, that is, independent of x. Even so, the model is a Gaussian Conditional (as opposed to a Gaussian Markov) Random Field because the observations x enter only via a nonparametric function $\mu(\cdot)$. Inspired by [43], we express the connectivity of our conditional Markov random field in terms of convolution kernels $\{f_k\}_{k=1}^K$. Unlike [43], we generalize the data term by learning regression trees that approximate linear functions of the latent variables, given the observed image x. These linear operations include finite differences and shifts.

More specifically, we posit

$$P = \sum_{k=1}^{K} F_k^T F_k \quad \text{and} \quad \mu(\mathrm{x}) = \sum_{k=1}^{K} F_k^T \mathrm{b}^k(\mathrm{x}; \mathcal{W}) \quad (7)$$

so that the MAP solution of (6) can be found by solving

$$\underset{\mathrm{z}}{\operatorname{argmin}} \sum_{k=1}^{K} \left\| F_k \cdot \mathrm{z} - \mathrm{b}^k(\mathrm{x}; \mathcal{W}) \right\|_F^2 \quad (8)$$

All matrices F_k are Toeplitz matrices representing convolution kernels in the spatial domain. As in [43], the first matrix F_1 is the identity matrix, while the others represent derivatives, see Fig. 2. In [43], $\mathrm{b}^1(\mathrm{x}; \mathcal{W}) = x$ and $\mathrm{b}^{k>1}(\mathrm{x}; \mathcal{W}) = 0$. In contrast, we learn the functions $\{\mathrm{b}^k\}_{k=1}^K$ (see section 2.2) to match differentiated and shifted label images, see Fig. 3.

In matrix notation, the objective function from (8) becomes

$$\left(\begin{bmatrix} F_1 \\ \vdots \\ F_K \end{bmatrix} \mathrm{z} - \begin{bmatrix} \mathrm{b}^1 \\ \vdots \\ \mathrm{b}^K \end{bmatrix}\right)^T \left(\begin{bmatrix} F_1 \\ \vdots \\ F_K \end{bmatrix} \mathrm{z} - \begin{bmatrix} \mathrm{b}^1 \\ \vdots \\ \mathrm{b}^K \end{bmatrix}\right) = (\mathbf{F}\mathrm{z} - \mathbf{b})^T (\mathbf{F}\mathrm{z} - \mathbf{b})$$

where $\mathbf{F} = [F_1; \ldots; F_K]$ stacks all convolution Toeplitz matrices column-wise and $\mathbf{b} = \left[\mathrm{b}^1; \ldots; \mathrm{b}^K\right]$ does so for the filter responses.

Eq. (8) can be solved in closed form:

$$\mathrm{z}(\mathrm{x}; \mathbf{b}) = \left(\mathbf{F}^T \mathbf{F}\right)^{-1} \mathbf{F}^T \mathbf{b} \quad (9)$$

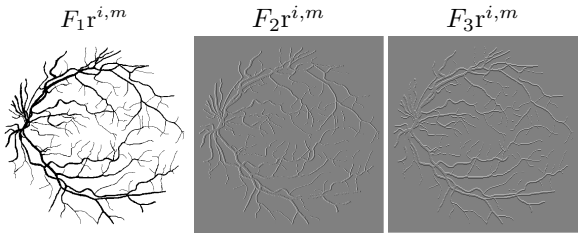

Figure 2. Example of the filter responses on $\mathrm{r}^{i,m} = \mathrm{y}^i$ to be learnt by $\{\mathrm{b}^k(\mathrm{x})\}_{k=1}^3$ as described in Eq. (12) for identity and horizontal- and vertical- derivative filters, respectively.

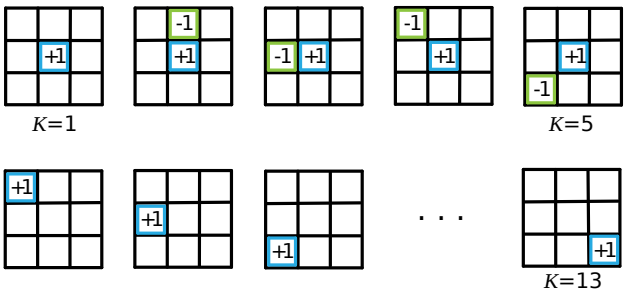

Figure 3. The filters used in the proposed model and grouped according to the neighborhood connectivity. Using only the identity filter amounts to the original gradient boosting framework ($K = 1$). It is here complemented with the first order vertical and horizontal finite differences ($K = 3$) plus diagonal finite differences ($K = 5$). In addition, adjacent labels can be queried by means of the shift operators ($K = 13$).

2.2. Learning

We now show how to obtain one of the weak learners $\mathrm{h}^m(\mathrm{x};\mathbf{b})$ in the form of a GCRF, which in turn depends on the response functions $\mathbf{b} = \{\mathrm{b}^{k,m}(\mathrm{x})\}_{k=1}^K$ at the m^{th} iteration:

$$\left\{\mathrm{h}^m, \{\mathrm{b}^{k,m}\}_{k=1}^K\right\} = \underset{\mathbf{b}(\cdot)\,\in\,\mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2} \left|\left|\mathrm{h}(\mathrm{x}^i;\mathbf{b}) - \mathrm{r}^{i,m}\right|\right|_F^2 \tag{10}$$

s.t.

$$\mathrm{h}(\mathrm{x}_i;\mathbf{b}) = \underset{\mathrm{z}}{\operatorname{argmin}}\ \mathrm{z}^T\mathbf{F}^T\mathbf{F}\mathrm{z} - 2\mathrm{z}^T\mathbf{F}^T\mathbf{b}(\mathrm{x}) + const.,$$

Note that Eq. (10) is a bilevel optimization problem; finding the best weak structured regressor requires also learning an estimated response function $\{\mathrm{b}^{k,m}\}_{k=1}^K$ that best predicts an underlying signal z close to the current gradient direction $\mathrm{r}^{i,m}$ at each sample x^i. Thanks to the closed-form solution from Eq. (9), Eq. (10) can be formulated as a direct optimization problem,

$$\{\mathrm{b}^{k,m}\}_{k=1}^K = \underset{\mathbf{b}(\cdot)\in\mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2} \left|\left|\left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{b} - \mathrm{r}^{i,m}\right|\right|_F^2. \tag{11}$$

This optimization problem can be rewritten in terms of a different quadratic objective function with the same minimizer:

$$\begin{aligned}\{\mathrm{b}^{k,m}\}_{k=1}^K &= \underset{\mathbf{b}(\cdot)\in\mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2} \left\|\mathbf{b} - \mathbf{F}\mathrm{r}^{i,m}\right\|^2 \qquad (12)\\ &= \underset{\mathbf{b}(\cdot)\in\mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2} \left\|\begin{bmatrix}\mathrm{b}^1\\ \vdots \\ \mathrm{b}^K\end{bmatrix} - \begin{bmatrix}F_1\\ \vdots \\ F_K\end{bmatrix}\mathrm{r}^{i,m}\right\|_F^2.\end{aligned}$$

Gradient descent based techniques have been used to learn parametric or non-parametric functions potentials for a GCRF [43, 44, 49, 14, 12, 15]. These techniques aim to find the global optimum of Eq. (11) with little or no restrictions on $\mathcal{B}$. Instead, following the boosting paradigm, we optimize only over a narrow class $\mathcal{B}$, in our case the class of shallow regression trees, to obtain weak learners. Limitations on tree depth are also used in [14, 12, 15] and are an alternative to early stopping of the gradient descent [43, 44, 49].

In practice, we approximately minimize the objective in Eq. (12) by training K shallow regression trees that, given an input image or its features x, try to approximate the filtered negative gradient loss images associated with the training images in the least-squares sense. This is a simple problem which, in addition, parallelizes naively over the K subproblems.

See Algorithm 1 for a summary of all steps[1].

[1]A helpful video that shows the temporal evolution of the prediction and the negative gradient direction at each iteration can be viewed at hci.iwr.uni-heidelderg.de/Staff/fdiego/SRGB/.

2.3. Inference

Given the learned real-valued weights $\{\alpha_t\}_{t=1}^M$ and the weak learners for each filter response $\{\{\mathrm{b}^{k,t}\}_{t=1}^M\}_{k=1}^K$, the global structured regression function φ becomes

$$\begin{aligned}\varphi^M(\mathrm{x}) &= \sum_{t=1}^{M} \alpha_t \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{b}^t(\mathrm{x}) \qquad (13)\\ &= \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\left(\sum_{t=1}^{M}\alpha_m\mathbf{b}^t(\mathrm{x})\right). \qquad (14)\end{aligned}$$

Computing Eq. (14) using direct methods is prohibitive due to the large number of variables (number of pixels). As a consequence, we solve the equivalent problem $\left(\mathbf{F}^T\mathbf{F}\right)\varphi^M(\mathrm{x}) = \mathbf{F}^T\left(\sum_{t=1}^M \alpha_t\mathbf{b}^t(\mathrm{x})\right)$ by conjugate gradient descent instead.

Note that at training time, the learning of the non-parametric filter responses entails solving M linear systems of equations; at test time, however, $\varphi^M(\mathrm{x})$ requires solving the linear system only once.

Algorithm 1 Training phase of Structured Regression Gradient Boosting (SRGB)

Input: Training samples and labels $\{(\mathrm{x}^i,\mathrm{y}^i)\}_{i=1}^N$, number of filters K, number of iterations M, shrinkage factor $0 < \gamma \leq 1$
Initialization: $\varphi^0(\cdot) = 0$
for $m = 1,\ldots,M$ **do**
  $\mathrm{r}^{i,m} = -\frac{\partial L(\mathrm{y}^i,\varphi^{m-1}(\mathrm{x}^i))}{\partial\varphi^{m-1}(\mathrm{x}^i)}$
  for $k = 1,\ldots,K$ **do**
    Train shallow regression trees $\mathrm{b}^{k,m}$ to match the kth filter response
    $\operatorname{argmin}_{\mathrm{b}(\cdot)} \sum_{i=1}^N \frac{1}{2}\left\|\mathrm{b}(\mathrm{x}^i) - F_k\mathrm{r}^{i,m}\right\|_F^2$
  end for
  $\mathrm{h}^m(\mathrm{x};\mathbf{b}) = \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{b}^m(\mathrm{x})$
  $\alpha_m = \operatorname{argmin}_\alpha \sum_{i=1}^N L\left(\mathrm{y}^i,\varphi^{m-1}(\cdot) + \alpha\mathrm{h}^m(\mathrm{x}^i)\right)$
  $\varphi^m(\cdot) = \varphi^{m-1}(\cdot) + \gamma\alpha_m\mathrm{h}^m(\cdot)$
end for
Output: $\varphi^M(\cdot)$

3. Experiments

The performance and the versatility of our method is evaluated on three different regression problems: blood vessel delineation in fundus images, depth estimation from single images, and Chinese character inpainting. Across the datasets, we provide comparison with baseline methods and with the state-of-the-art on those datasets.

Method	Prec.	Recall	F	AUC
CS [31]	78.81	74.74	76.72	84.12
FC-CRF [28]	79.10	78.08	78.55	–
Kernel Boost [2, 3]	81.10	79.75	80.42	88.84
SE [8, 7]	68.22	65.33	66.74	70.70
N^4 [10]	80.41	**80.76**	80.58	88.93
Learning Boost [11]	79.57	79.47	79.51	–
NN Projections [38]	81.28	79.95	80.61	84.97
SRGB	**81.67**	80.16	**80.91**	**89.17**

Table 1. Vessel segmentation. Comparison to the state of the art on the DRIVE dataset.

F-measure	iter 0		iter 1	iter 2	iter 3
	sep.	joint			
$K = 1$	78.82		80.33	80.62	80.81
$K = 3$	79.53	**78.93**	80.59	80.80	80.86
$K = 5$	**79.70**	78.17	80.57	80.74	80.75
$K = 13$	79.54	77.93	**80.68**	**80.89**	**80.91**

Table 2. Vessel segmentation. Performance for various choices of K (and thus implicitly for the connectedness of the Gaussian Conditional Random Field) and for different number of autocontext stages. In the first stage, we differentiate between learning independently a nonparametric regression tree for each of the filter responses and learning jointly the influence among filter responses.

3.1. Vessel Segmentation

The first problem is the segmentation of blood vessels in retinal scans. We test our approach on the DRIVE dataset [41], which contains 20 training images and 20 test images of size 565×584 as well as the corresponding manual vessel segmentation and ROI masks.

We compare our approach with the latest and state-of-the-art methods: CS [31], Fully-Connected CRF [28], SE [8, 7], KernelBoost [2, 3], N^4-fields [10], Learning Boost [11] and NN-projections [38]. The latter two approaches [38, 11] refine and enhance the segmentation given a base segmentation algorithm, in this case [39] and [3], respectively. Earlier, the features were learned discriminatively using the Gradient Boosting framework [2, 3] or a neural network [10]. Following [8, 7], we use hand-crafted features as implemented in [40] that comprise smoothing filters, edge detectors and eigenvalues of the structure tensor at different scales (49 features in total). Finally, following KernelBoost [2], we implement the autocontext [48] idea by training a cascade of 4 sequential Structured Regression Gradient Boosting predictors. Each predictor receives all the original features plus the output of the previous stage as input. The number of boosting iterations was set to $M = 500$, the tree depth is limited to 3 levels for each weak filter learner, the regularizing shrinkage factor (see Algorithm 1) is set to $\gamma = 0.1$ and the log-loss is used. All of the above are standard choices for gradient boosting.

Precision/recall curves for the baseline Gradient Boosting framework and the proposed method at different levels of the cascade are shown in Fig. 4; while Table 2 shows the F–measure obtained for different relation among latent variables and for different number of cascade iterations. There is a clear advantage over the baseline at the first stage when neighboring information is considered. The increase in performance is reduced at the last stage, although our proposed approach requires fewer cascade stages to achieve a given performance level.

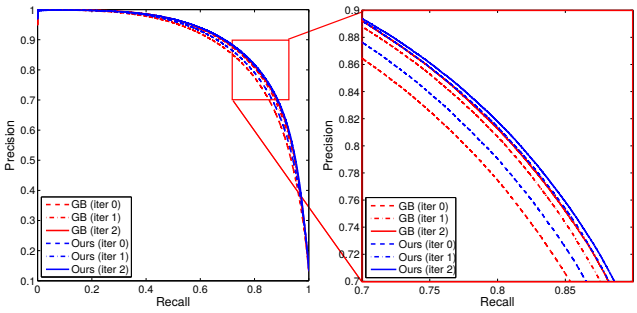

Figure 4. Results for the DRIVE dataset [41] in the form of the recall/precision curves. Our approach outperforms the baseline in the last stages of the cascade.

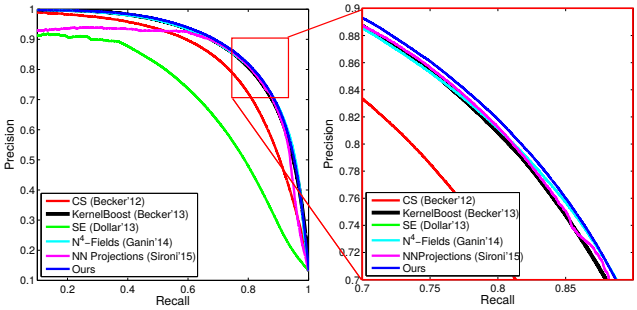

Figure 5. Results for the DRIVE dataset [41] in the form of the recall/precision curves. Our approach is slightly above the performance of the current state-of-the-art methods by Ganin *et al*. [10], Becker *et al*. [3] and Sironi *et al*. [38], and much better than [8] and [31].

Table 1 and Fig. 5 show the F-measure and the precision/recall curves obtained with the different methods, respectively. Our approach is slightly better than the state-

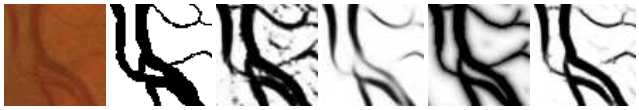

Figure 6. Example of vessel segmentation on a region with complex topology. From left to right: raw image, ground truth; Kernel Boost [2, 3], N^4 Fields [10], NN-Projections [38] and Structured Regression Gradient Boosting. Arguably some of these predictions are more accurate than the ground truth.

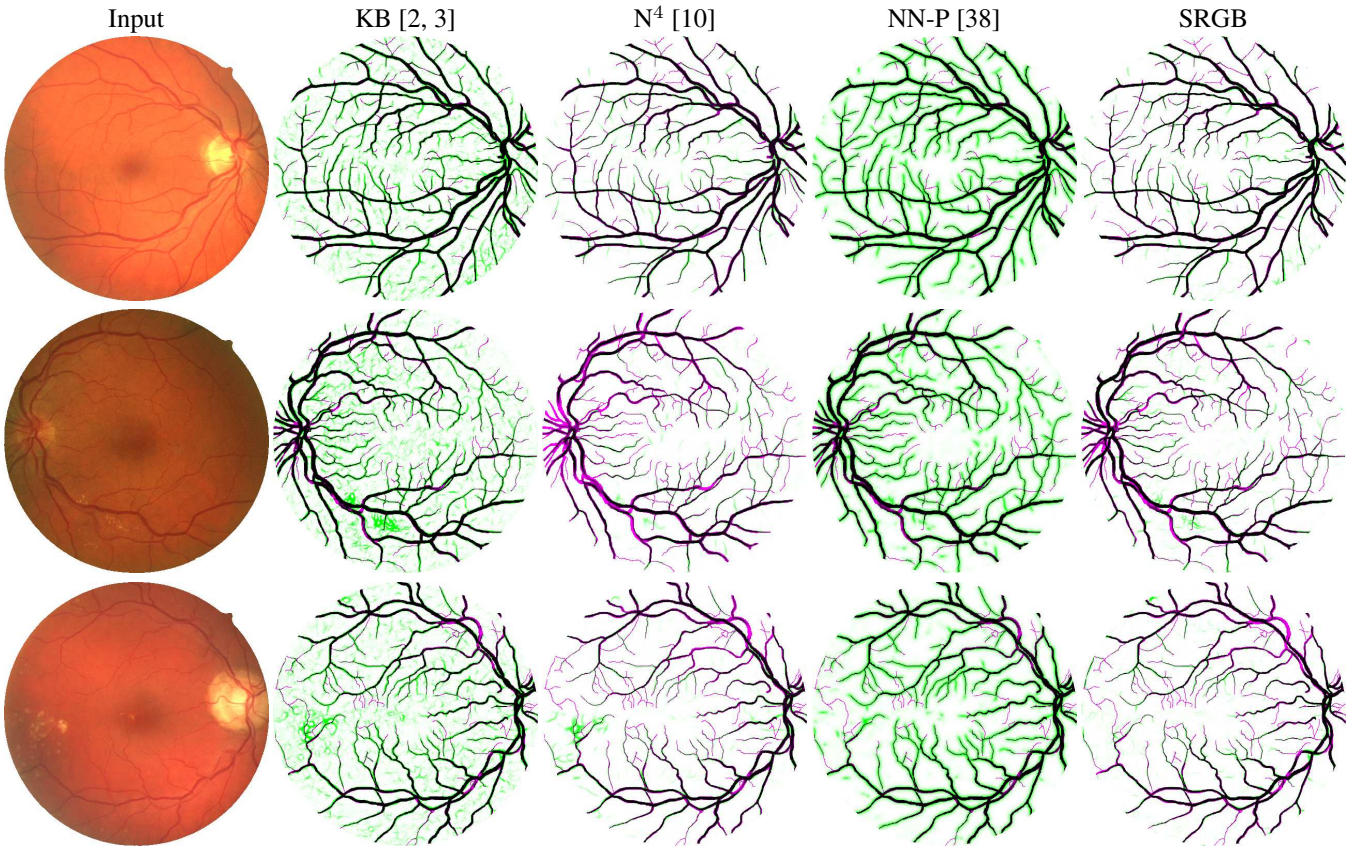

Figure 7. Examples of vessel segmentation on the DRIVE dataset (best viewed on screen). From left to right: original image, Kernel Boost [2, 3], N^4 Fields [10], NN-Projections [38] and the method presented here. True positives are shown in black, false positives in green and false negatives in magenta. The human-generated ground truth is somewhat subjective, but is the only one available and is the standard by which all methods are measured. We closely emulate the human expert while reducing false positives.

of-the-art N^4 fields [10], NN-Projection [38] and KernelBoost [3], and outperforms the other competitors with only a single stage of the cascade because the first-round weak structured learners aim to learn coarse and rough details and the later ones aim to accurately learn finer details. Some qualitative results are shown in Fig. 7. Similarly to [38], Fig. 6 shows the results on the same particularly complex region of a test image, with several thin junctions and low contrast, for N^4 fields [10], NN-Projection [38] and KernelBoost [3]. Our approach is able to reconstruct correctly most of the topology of the blood network except for the junction of a tiny blurred vessel. Moreover, as can be seen from Fig. 6, our method predicts even better vessel locations than the ground truth in some part of the image and with a very high confidence.

3.2. Depth Estimation

The second problem is the depth estimation from single image. We test our approach on the Make3D dataset [34, 35, 36], which contains 400 training images and 134 test images as well as the corresponding depth (we use the standard training/test split provided with the dataset). Following [16], we resize the images to 345×460 pixels before training (maintaining the aspect ratio of the input images).

We rely on the main idea of depth transfer [16] based on non-parametric learning [21] to compute the features. The features consists on finding the top-N best candidate images in a similar database of the input image, and on using SIFT Flow [22] to warp these images as well as the depths to the query image. Moreover, we extend our features by adding another "coarse" depth map obtained from DCNF [23] without any inpainting enhancement. Finally, we train just a single Structured Regression Gradient Boosting, and use a square loss as an empirical risk. The number of boosting iterations M was set to 50 and the shrinkage is set to 0.1 but the tree depth is limited to 5 levels for each weak filter learner.

In Table 3 we compare our approach with different set of convolutional kernels and with state-of-the-art methods. From the table, we conclude that exploiting smooth prior as $4-$ and $8-$neighborhood increases the baseline performance and it outperforms with a large margin if more non-

Method	Error		
	rel	log10	rms
Depth MRF [34]	0.530	0.198	-
Make3D [36]	0.370	0.187	-
Semantic Labelling [20]	0.379	0.148	-
Laplace CRF [1]	0.362	0.168	15.2
Depth Transfer [16]	0.361	0.148	15.10
DiscreteContinuous CRF [25]	0.338	0.134	12.60
DCNF [23]	0.307	0.125	12.89
DCNF-FCSP [24]	**0.305**	0.120	13.24
SRGB ($K = 1$)	0.309	0.119	11.76
SRGB($K = 3$)	0.309	0.119	11.65
SRGB ($K = 27$)	0.315	**0.118**	**11.48**

Table 3. Depth Estimation. State-of-the-art and baseline comparisons on the Make3D dataset.

local information is used to predict the value of a current pixel. Moreover this framework allows to infer the depth by a non-linear combination of multiple observations as features and outperforms the state-of-the-art. Some examples of qualitative evaluations are shown in Fig. 8. It is shown that only Gradient Boosting model gives rather coarse predictions, but our model yields much better visualizations by adding smoothness term and more nonlocal prior.

3.3. Chinese character inpainting

The third problem is learning calligraphy properties for the reconstruction of the occluded parts of handwritten Chinese characters from the KAIST Hanja2 database (Fig. 9). We used the original 300 training images and 100 test images in [27]. Each character is occluded by a centered grey box of varying size. Following [27], the accuracy is measured on a dataset with small occlusions, and the predictions are visualized on images with a larger occlusion area. We differ from the DTF [27] model slightly in terms of features and neighborhood. Instead of looking at most 80 pixels away, we restricted our search area to only 31 pixels away, but also including the difference between these two pixels. Moreover, we restricted the relation among latent variables to 4- and 8-neighbor connections, in this case $K = 3$ and $K = 5$ respectively. Finally, we train just a single Structured Regression Gradient Boosting, and use a log loss as a empirical risk. The number of boosting iterations M was set to 100 and the shrinkage is set to 0.1 but the tree depth is again limited to 3 levels for each weak filter learner.

The results are shown in Table 4. We include the baseline decision tree result (DT) of [27], a tree ensemble result of 10 trees (RF) from [32, 17], the MRF and DTF results taken from [27], the Regression Tree Field (RTF) approaches from [14], the convex quadratic relaxation approach (QP-M3N) of [13], and the structured local predictors (SLP) [32] and structured labels in Random Forests (SLRF). Additionally, we compare the classification results for Gradient Boosting and our proposed structured prediction. Our baseline method outperforms most of sophisticated state-of-the-art methods; thanks to learning how to perform the gradient descent for reconstructing the occluded parts. Hence, our proposed method that learns *jointly* the interaction among neighboring pixels is very competitive, and achieves the best result on this task. Following the works in [27, 14, 32, 17], Fig. 9 shows the some qualitative results obtained on the large occlusion dataset.

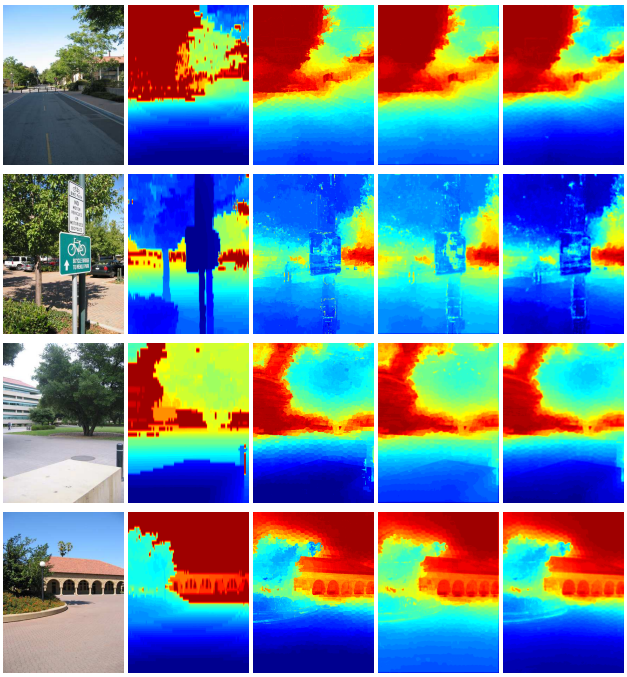

Figure 8. Examples of depth predictions on the Make3D dataset (Best viewed on screen). From left to right: original image, ground truth, gradient boosting baseline ($K = 1$), our gradient boosting with pairwise connectivity ($K = 3$) and our approach with pairwise connectivity and with the data term computed from a fully connected 5×5 neighborhood ($K = 27$). The gradient boosting gives rather coarse prediction mainly due to the coarse depth estimations used as features. In contrast, our full model yields much better predictions.

3.4. Practicalities: choice of parameters

For vessel segmentation, we used the parameters published in [2].

For the other experiments, the number of iterations M was increased when the loss on the training set was found to be high (indicative of underfitting). The maximum depth of the regression trees was chosen depending on the dynamic range of the desired output. For targets in $[-1, 1]$ a depth of 3 was used, and for a larger range a depth of 5.

DT (avg) [17]	RF [17]	MRF [27]	DTF [27]	RTF_{1D} [14]	RTF_{2D} [14]	SLP [32]	QP_{M3N} [13]	SLRF [17]	SRGB ($K = 1$)	SRGB ($K = 3$)	SRGB ($K = 5$)
68.52	74.95	75.18	76.01	76.39	77.55	78.07	79.36	78.09	77.66	79.37	**79.87**

Table 4. Chinese characters: accuracy for inpainting of small occlusions.

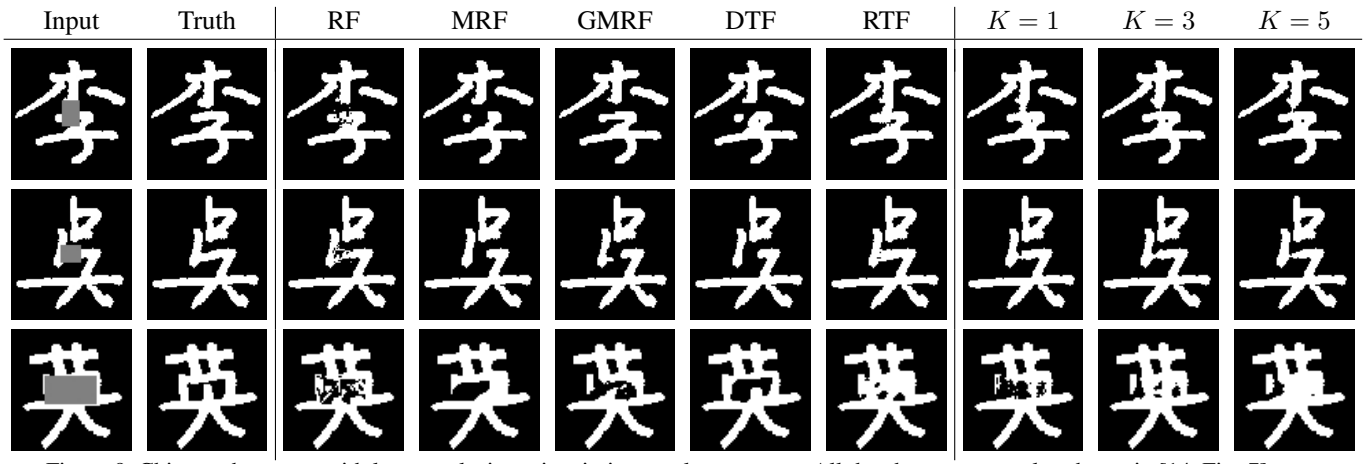

Figure 9. Chinese characters with large occlusions: inpainting result on test set. All the characters are also shown in [14, Fig. 7].

4. Conclusions and Outlook

We propose a structured regression gradient boosting method. Analogous to gradient boosting, a non-linear structured output regression is obtained as a combination of a set of weak structured learners, while the original formulation is kept in the framework as a specific case. Inspired by [43, 12], we use a GCRF as a weak structured learner, whose parameters are learnt discriminatively by means of nonparametric regression trees. Our proposed approach is comparable to and sometimes exceeds the state-of-the-art even with less feature tuning for three challenging benchmarks. We also observe that the proposed approach has improved performance over gradient boosting, demonstrating the usefulness of this nonlinear *structured* regression method.

Future work includes generalization to multiclass predictions, automated selection of the most useful filters F, learning of features and letting precision matrix P depend on the input.

References

[1] D. Batra and A. Saxena. Learning the right model: Efficient max-margin learning in laplacian crfs. In *CVPR*, 2012.

[2] C. J. Becker, R. Rigamonti, V. Lepetit, and P. Fua. KernelBoost: Supervised Learning of Image Features For Classification. Technical report, 2013.

[3] C. J. Becker, R. Rigamonti, V. Lepetit, and P. Fua. Supervised Feature Learning for Curvilinear Structure Segmentation. In *MICCAI*, 2013.

[4] T. Chen, S. Singh, B. Taskar, and C. Guestrin. Efficient second-order gradient boosting for conditional random fields. In *AISTATS*, 2015.

[5] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8, 2002.

[6] T. G. Dietterich, G. Hao, and A. Ashenfelter. Gradient Tree Boosting for Training Conditional Random Fields.

[7] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[8] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[9] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 2000.

[10] Y. Ganin and V. Lempitsky. N4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*, 2014.

[11] L. Gu and L. Cheng. Learning to boost filamentary structure segmentation. In *ICCV*, 2015.

[12] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. In *ECCV*, 2012.

[13] J. Jancsary, S. Nowozin, and C. Rother. Learning convex qp relaxations for structured prediction. In *ICML*, 2013.

[14] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression tree fields: An efficient, non-parametric approach to image labeling problems. In *CVPR*, 2012.

[15] O. Kahler and I. Reid. Efficient 3d scene labeling using fields of trees. In *ICCV*, 2013.

[16] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. 2014.

[17] P. Kontschieder, S. Rota Bulo, M. Pelillo, and H. Bischof. Structured labels in random forests for semantic labelling and object detection. *PAMI*, 36, 2014.

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

[19] L. Liao, T. Choudhury, D. Fox, and H. Kautz. Training conditional random fields using virtual evidence boosting. In *IJCAI*. 2007.

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

[21] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 2011.

[22] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2011.

[23] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

[24] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *PAMI*, 2015.

[25] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. 2014.

[26] S. Nowozin, P. V. Gehler, J. Jancsary, and C. H. Lampert. *Advanced Structured Prediction*. 2014.

[27] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *ICCV*, 2011.

[28] J. I. Orlando and M. Blaschko. Learning fully-connected CRFs for blood vessel segmentation in retinal images. In *MICCAI*. 2014.

[29] C. Parker. *Structured Gradient Boosting*. PhD thesis, 2007.

[30] N. Ratliff, D. Bradley, J. A. D. Bagnell, and J. Chestnutt. Boosting Structured Prediction for Imitation Learning. In *NIPS*, 2007.

[31] R. Rigamonti and V. Lepetit. Accurate and Efficient Linear Structure Segmentation by Leveraging Ad Hoc Features with Learned Filters. In *MICCAI*, 2012.

[32] S. Rota Bulo, P. Kontschieder, M. Pelillo, and H. Bischof. Structured local predictors for image labelling. In *CVPR*, 2012.

[33] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. Monographs on Statistics and Applied Probability. 2005.

[34] A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[35] A. Saxena, M. Sun, and A. Ng. Learning 3-d scene structure from a single still image. In *ICCV workshop on 3dRR*, 2007.

[36] A. Saxena, M. Sun, and A. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 2009.

[37] C. Shen, G. Lin, and A. van den Hengel. StructBoost: Boosting methods for predicting structured output variables. *PAMI*, 2014.

[38] A. Sironi, V. Lepetit, and P. Fua. Projection onto the Manifold of Elongated Structures for Accurate Extraction. In *ICCV*, 2015.

[39] A. Sironi, E. Tretken, V. Lepetit, and P. Fua. Multiscale Centerline Detection. *PAMI*, 2015.

[40] C. Sommer, C. Strähle, U. Köthe, and F. A. Hamprecht. ilastik: Interactive learning and segmentation toolkit. In *Eighth IEEE International Symposium on Biomedical Imaging (ISBI 2011). Proceedings*, pages 230–233, 2011. 1.

[41] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE TMI*, 2004.

[42] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. In *ECCV*, 2008.

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

[44] M. F. Tappen, K. G. G. Samuel, C. V. Dean, and D. M. Lyle. The logistic random field a convenient graphical model for learning parameters for mrf-based labeling. In *CVPR*, 2008.

[45] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*. 2004.

[46] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*. 2005.

[47] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005.

[48] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *PAMI*, 2010.

[49] J. Zhu, K. Samuel, S. Masood, and M. Tappen. Learning to recognize shadows in monochromatic natural images. In *CVPR*, 2010.