This ICCV2013 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Active MAP Inference in CRFs for Efficient Semantic Segmentation

Gemma Roig[1] * Xavier Boix[1] *
Roderick de Nijs[2] Sebastian Ramos[3] Kolja Kühnlenz[2] Luc Van Gool[1,4]
[1]ETH Zürich, Switzerland [2]TU Munchen, Germany [3]CVC Barcelona, Spain [4]KU Leuven, Belgium
* *Both first authors contributed equally.* {boxavier,gemmar}@vision.ee.ethz.ch

Abstract

Most MAP inference algorithms for CRFs optimize an energy function knowing all the potentials. In this paper, we focus on CRFs where the computational cost of instantiating the potentials is orders of magnitude higher than MAP inference. This is often the case in semantic image segmentation, where most potentials are instantiated by slow classifiers fed with costly features. We introduce Active MAP inference 1) to on-the-fly *select a subset of potentials to be instantiated in the energy function, leaving the rest of the parameters of the potentials unknown, and 2) to estimate the MAP labeling from such incomplete energy function. Results for semantic segmentation benchmarks, namely PASCAL VOC 2010 [5] and MSRC-21 [19], show that Active MAP inference achieves similar levels of accuracy but with major efficiency gains.*

1. Introduction

In many state-of-the-art methods for semantic segmentation, contextual information plays a central role. A successful trend has been to encode the contextual constraints with a Conditional Random Field (CRF) [11], by modeling the interactions between different regions and scales of the image. Most methods use sophisticated potentials between different neighboring regions [7, 21], and the state-of-the-art has been boosted with the use of high-order potentials in hierarchical CRFs [2, 9, 17].

Another common way to include contextual information has been to extend image descriptors with contextual cues [6, 8, 15], or also, combining semantic classifiers fed from different contextual features [4, 13, 14]. It is a remarkable feat the balance struck between accuracy and efficiency by the semantic texton forests of Shotton *et al.* [19]. The good performance exhibited by many methods that do not benefit from introducing context to a CRF, lead Lucchi *et al.* [12] ask the provocative question: 'Are spatial and global constraints really necessary for segmentation?' From the experimental results, they conclude that the CRF structures boost performance when the features only encode local information, whereas the further gain is very little when the features already encode contextual information. This begs the question whether we can really benefit from CRFs in semantic segmentation when using such powerful features that already encode context.

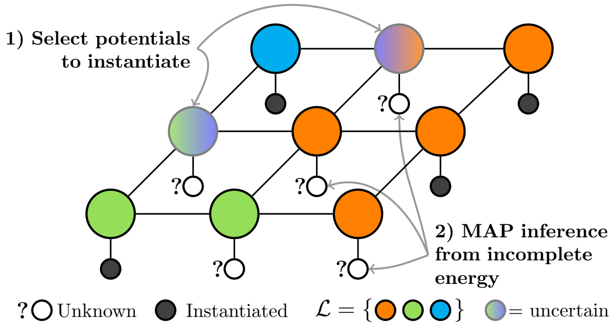


Figure 1. *Active MAP inference (best seen in color).* Example of CRF with unknown unary potentials. Active MAP selects the potentials to instantiate that maximize the expected reward. Also, it estimates the MAP labeling from the incomplete energy function.

We present a novel use of CRFs for semantic segmentation. We exploit CRFs to estimate the semantic labeling without computing the descriptors and classifiers everywhere in the image. Given a budget of time, it decides which potentials to compute. In doing so, it dramatically reduces the computational complexity of the whole pipeline. This is because the computational burden of instantiating the potentials that extract descriptors and apply classifiers, which can be much higher than MAP inference for most of the energy functions in the literature [2, 6, 12].

We introduce a relation between CRFs with some unknown unary potentials, which correspond to the features and classifiers that we do not compute, and the Perturb-and-MAP (PM) random field model [16]. We build our MAP inference algorithm - coined Active MAP inference - based on this finding. We use the term 'active' because during inference it selects which potentials to instantiate *on-the-fly*. This stands in contrast to previous MAP inference methods, which first execute the features/classifiers that instantiate

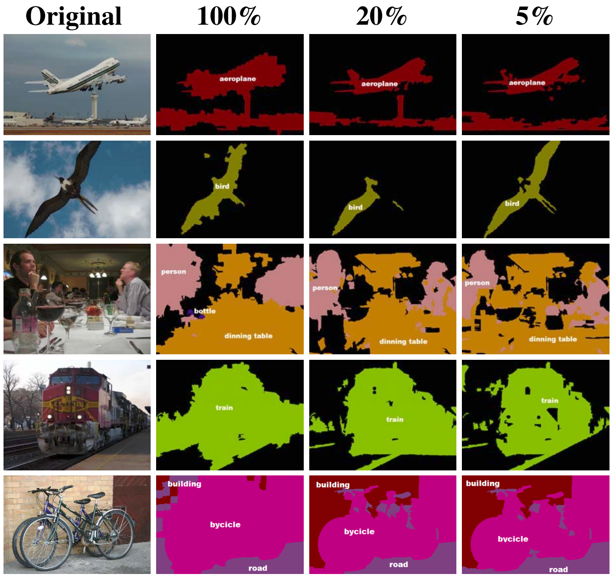


Figure 2. *Examples of segmented images on VOC10 and MSRC-21.* Active MAP inference using observing different amounts of unary potentials. Results are obtained by selecting the unary potentials with the expected labeling change.

the CRF, and then run the MAP-CRF inference. Surprisingly, seeing the instantiation of the CRF energy function and MAP-CRF inference as two joint steps received little attention in the community.

In a serie of experiments, we show that active MAP inference successfully exploits spatial consistency to avoid evaluating the classifiers and features everywhere. It obtains comparable results to instantiating all the potentials in the CRF for the PASCAL VOC 2010 segmentation challenge [5] and for the MSRC-21 dataset [19], but with major efficiency gains. In Fig. 1 we illustrate some results on semantic segmentation obtained with active MAP inference.

2. Active MAP Inference in CRFs

This section describes the approach for active MAP inference. Its formulation uses a CRF to model the probability density distribution expressing the likeliness of a certain labeling. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph that represents such distribution, and $\mathbf{X}$ the set of random variables or nodes of the graph. The elements of $\mathcal{V}$ are indices of the nodes, *i.e.* $\mathbf{X} = \{X_i\}$ in which $i \in \mathcal{V}$, and the elements of $\mathcal{E}$ are the indices of the undirected edges of the graph. We denote an instance of the random variables as $\mathbf{x} = \{x_i\}$, where x_i takes a value from a set of discrete labels $\mathcal{L}$. Thus, $\mathbf{x} \in \mathcal{L}^N$, with N the cardinality of $\mathcal{V}$.

We denote $P(\mathbf{x}|\boldsymbol{\theta})$ as the probability density distribution of a labeling modeled with the graph $\mathcal{G}$. According to the Hammersley-Clifford theorem (*c.f.* [10]), the probability density that satisfies the Markov properties with respect to the graph $\mathcal{G}$ is a Gibbs distribution. Thus, $P(\mathbf{x}|\boldsymbol{\theta})$ can be written as the normalized negative exponential of an energy function $E_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})$, in which $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(x), \dots, \phi_M(x))^T$ is the vector of potentials, or the so-called sufficient statistics, and $\boldsymbol{\theta} \in \mathbb{R}^M$ are the parameters of the potentials. We use the *canonical over-complete representation*, in which $\{\phi_i(\mathbf{x})\}$ are built using indicator functions that allow us to express the energy function as such linear combination of the potentials (*c.f.* [23]). The most probable state $\mathbf{x}^\star$ is obtained by inferring the Maximum a Posteriori (MAP) of $P(\mathbf{x}|\boldsymbol{\theta})$, or equivalently by minimizing the energy, *i.e.* $\mathbf{x}^\star = \arg\min_{\mathbf{x} \in \mathcal{L}^N} \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})$. As usual, we categorize the potentials of the energy function depending on the number of random variables that they involve: unary and pairwise.

In the case of semantic segmentation, there is a node defined for each pixel or superpixel in the image. The parameters of $\boldsymbol{\theta}$ related to the unary potentials are typically the result of evaluating classifiers fed with features extracted from the image. The pairwise and high-order potentials use some *a priori* assumptions like the smoothness of the labeling. It is important to note that the instantiation of $\boldsymbol{\theta}$ might be orders of magnitude more computationally expensive than MAP inference. Usually, state-of-the-art methods for semantic segmentation use features and classifiers that take minutes to compute for a single image [2, 6, 12].

At testing phase, the common way to proceed is to instantiate $\boldsymbol{\theta}$, and then to run an off-the-shelf MAP inference algorithm to obtain the most probable labeling. Active MAP inference aims at estimating $\mathbf{x}^\star$ with only a subset of the elements of $\boldsymbol{\theta}$, $\{\theta^j\}$, which is selected by the algorithm. The computational gain comes from not computing all classifiers and features needed to fully instantiate $\boldsymbol{\theta}$. Even though we do not have the complete energy function anymore because part of $\boldsymbol{\theta}$ is unknown, we will show in the sequel that we can still estimate $\mathbf{x}^\star$. We define $\boldsymbol{\delta} \in \{0, 1\}^M$, with the purpose of introducing the concept of selected parameters in our notation, *i.e.* it works as an indicator function. When the element j of the vector $\boldsymbol{\theta}$, *i.e.* θ^j, is not computed, then, δ^j is zero, and if the parameter is computed, then δ^j equals 1. This is

$$\theta^j_{\delta^j} = \begin{cases} \theta^j & \text{if } \delta^j = 1 \\ \text{unknown} & \text{otherwise} \end{cases} . \tag{1}$$

Note that with this notation we can still easily express the initial formulation that instantiates all parameters, using $\boldsymbol{\delta} = \mathbf{1}$ and $\boldsymbol{\theta}_{\mathbf{1}}$, where $\mathbf{1}$ is a vector of ones.

With missing parameters, the energy function does not represent the initial labeling problem anymore. It would be wrong to replace the unknown parameters by 0, or any value indicating that 'the potential is missing'. There is no guarantee that, in doing so, the new energy function would assign energy values similar to the ones given by the complete energy.

General Overview. Given a budget of time, Active MAP inference instantiates a subset of the potentials ($\boldsymbol{\delta}$), and only with them, it computes the complete MAP labeling ($\mathbf{x}^\star$). In the following section, we introduce Perturb-and-MAP, as we use this mathematical tool in the rest of the paper. In section 4, we introduce the estimation of $\mathbf{x}^\star$ when $\boldsymbol{\delta}$ is given, and in section 5, we introduce the algorithm to determine $\boldsymbol{\delta}$. Finally, we show results for the application of semantic image segmentation, where we save the cost of instantiating all the unary potentials. Active MAP inference is more general and can also be applied in many other applications.

3. Preliminaries: Perturb-and-MAP

Generating samples from CRFs is unusual in computer vision. For most problems, sampling over the discrete space of the CRF is prohibitive due to the complexity of these spaces. Recently, Papandreou and Yuille introduced the PM random field [16], which is a model that allows for generating samples, built around the effective MAP inference algorithms in CRF. In a follow-up paper, Tarlow *et al.* [20] extended this idea to a more broader set of model.

PM is based on injecting noise in the energy function to perturb it, and then, it calculates the frequency that labelings are the MAP of the perturbed energy. Let $\boldsymbol{\epsilon} \in \mathbb{R}^M$ be the random variable that it is used to perturb the parameters of the energy function, and let $f_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})$ be the probability density of $\boldsymbol{\epsilon}$. We denote the perturbed parameters of the energy as $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + \boldsymbol{\epsilon}$. For each perturbed $\tilde{\boldsymbol{\theta}}$, we can infer a MAP labeling. The different $\tilde{\boldsymbol{\theta}}$s that yield the same MAP labeling $\mathbf{x}$, can be grouped together. We use $\mathcal{P}_{\mathbf{x}}$ to denote such set of $\tilde{\boldsymbol{\theta}}$s,

$$\mathcal{P}_{\mathbf{x}} = \left\{ \tilde{\boldsymbol{\theta}} \in \mathbb{R}^M | \mathbf{x} = \arg\min_{\mathbf{x}' \in \mathcal{L}^N} \tilde{\boldsymbol{\theta}}^T \boldsymbol{\phi}(\mathbf{x}') \right\}. \qquad (2)$$

Analogously, we can define the set of perturbations $\boldsymbol{\epsilon}$, that yields the labeling $\mathbf{x}$ when doing MAP inference. We denote this set as $\mathcal{P}_{\mathbf{x}} - \boldsymbol{\theta}$, and it is $\left\{\boldsymbol{\epsilon} \in \mathbb{R}^M | \mathbf{x} = \arg\min_{\mathbf{x}' \in \mathcal{L}^N} (\boldsymbol{\theta} + \boldsymbol{\epsilon})^T \boldsymbol{\phi}(\mathbf{x}')\right\}$. PM assigns a probability to $\mathbf{x}$ equal to the probability of drawing a perturbation $\boldsymbol{\epsilon}$ that belongs to the set $\mathcal{P}_{\mathbf{x}} - \boldsymbol{\theta}$. Thus, the PM distribution is

$$f_{PM}(\mathbf{x}|\boldsymbol{\theta}) = \int_{\mathcal{P}_{\mathbf{x}} - \boldsymbol{\theta}} f_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}, \qquad (3)$$

Intuitively, the PM calculates how frequent is that a labeling $\mathbf{x}$ is the MAP labeling, when injecting noise to the energy function. Even though calculating the exact value of $f_{PM}(\mathbf{x}; \boldsymbol{\theta})$ might be not feasible for most practical cases, note that we can easily draw samples from a PM distribution by simply doing MAP inference on a perturbed energy. For a complete explanation of the PM random field we refer the reader to the paper [16].

4. MAP Inference for Incomplete Energies

This section aims at estimating the labeling from the incomplete energy function. We assume that $\boldsymbol{\delta}$ is given, and the potentials indicated by $\boldsymbol{\delta}$ have been instantiated.

4.1. Relation to Perturb-and-MAP

Rather than filling in the energy function by inventing the unknown parameters or setting them to a learned constant value, we use $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ to model them. $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ is the probability that the parameters of the potentials take the values $\boldsymbol{\theta}$ given $\boldsymbol{\theta}_{\boldsymbol{\delta}}$. The CRF models the probability of the labeling, but it does not directly model $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$. In order to alleviate the lack of an exact expression for $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$, we use a model to approximate it, referred to as $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$, where $\boldsymbol{\pi}$ are the parameters of the model. The definition of this model is open and adaptable to each problem. We specify $f_{\boldsymbol{\theta}}$ and $\boldsymbol{\pi}$ in the subsequent section.

Changing $\boldsymbol{\theta}$ in the energy function produces different MAP labelings, $\mathbf{x}^\star$. Therefore, $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ induces a probability on $\mathbf{x}^\star$. We use $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ to define such probability on $\mathbf{x}^\star$, *i.e.* the probability that $\mathbf{x}$ is the MAP labeling. It can be computed as

$$\int_{\mathbb{R}^M} \mathbf{I}\left[\mathbf{x} = \arg\min_{\mathbf{x}' \in \mathcal{L}^N} E_{\boldsymbol{\theta}}(\mathbf{x}')\right] P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}}) d\boldsymbol{\theta}, \qquad (4)$$

where $\mathbf{I}[\cdot]$ is the indicator function. Eq. (4) can be seen as a natural way to calculate $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$, since it accumulates the probability density of $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ with $\boldsymbol{\theta}$ yielding the minimum energy labeling equal to $\mathbf{x}$. The integral explores all complete energy functions, $E_{\boldsymbol{\theta}}(\mathbf{x})$, and for each of them, it checks whether the MAP labeling is $\mathbf{x}$ or not. In case it is equal to $\mathbf{x}$, the corresponding probability density of $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ is accumulated into the final probability.

Deriving the exact $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ is computationally intractable, because of the number and complexity of the constraints needed to define $E_{\boldsymbol{\theta}}$. Fortunately, it can be shown that $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ is indeed a PM random field, from which we can easily draw samples. We state it formally in the following proposition.

Proposition 1. *Let $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}}) = f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$, and $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$ has mean equal to $\boldsymbol{\mu} \in \mathbb{R}^M$. Let $f_{PM}(\mathbf{x}|\boldsymbol{\delta}, \boldsymbol{\mu})$ be the density distribution of a PM model with energy $E_{\boldsymbol{\mu}}(\mathbf{x})$,* i.e. *the energy with parameter $\boldsymbol{\mu} \in \mathbb{R}^M$, and the perturbations are drawn from $\boldsymbol{\epsilon} \sim f_{\boldsymbol{\theta}}(\boldsymbol{\epsilon} + \boldsymbol{\mu}|\boldsymbol{\delta}, \boldsymbol{\pi})$. Then,*

$$P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}}) = f_{PM}(\mathbf{x}|\boldsymbol{\delta}, \boldsymbol{\mu}). \qquad (5)$$

The proof is in the Supplementary Material. Observe that the density distribution of the PM model in Prop. 1 is

$$f_{PM}(\mathbf{x}|\boldsymbol{\delta}, \boldsymbol{\mu}) = \int_{\mathcal{P}_{\mathbf{x}} - \boldsymbol{\mu}} f_{\boldsymbol{\theta}}(\boldsymbol{\epsilon} + \boldsymbol{\mu}|\boldsymbol{\delta}, \boldsymbol{\pi}) d\boldsymbol{\epsilon}, \qquad (6)$$

where $\mathcal{P}_{\mathbf{x}} - \boldsymbol{\mu}$ is the set of $\boldsymbol{\epsilon} \in \mathbb{R}^M$ such that $\mathbf{x}$ minimizes the energy function $E_{(\boldsymbol{\mu}+\boldsymbol{\epsilon})}$ (see Eq. (2)). Note also that we draw $\boldsymbol{\epsilon}$ from $f_{\boldsymbol{\theta}}(\boldsymbol{\epsilon} + \boldsymbol{\mu}|\boldsymbol{\delta}, \boldsymbol{\pi})$, which is $f_{\boldsymbol{\theta}}$ centered at $\mathbf{0}$. Prop. 1 shows that this PM distribution reproduces the definition of $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ in Eq. (4). To obtain samples of $\mathbf{x}^\star$ in practice, we simply perturb $\boldsymbol{\mu}$ using $\boldsymbol{\epsilon}$, and then, we apply MAP inference to $E_{(\boldsymbol{\mu}+\boldsymbol{\epsilon})}(\mathbf{x})$.

Note that Prop. 1 is valid for any $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$. Yet, the key assumption in Prop. 1 is $P(\boldsymbol{\theta}|\boldsymbol{\theta}_{\boldsymbol{\delta}}) = f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$, which presupposes an underlying model for the known and unknown θ. This is addressed in the following.

4.2. Model of the Missing Parameters

We use a simple collection of independent Gaussian variables to define $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$. The parameters for this model are the mean and the standard deviation, referred to as $\boldsymbol{\mu} \in \mathbb{R}^M$ and $\boldsymbol{\sigma} \in \mathbb{R}^M$ respectively, where for notation simplicity $\boldsymbol{\pi}$ indicates both $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. We use the standard Gaussian distribution due to its simplicity and its well-known properties. Specifically, we define $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{\delta}, \boldsymbol{\pi})$ such that, if the parameter of the potential is unknown ($\delta^i = 0$), it is a univariate Gaussian distribution, centered at μ^i and deviation σ^i. Otherwise it is consistent with the instantiated potential, $f_\theta(\theta^i|\delta^i = 1, \pi^i) = \mathbf{I}[\theta^i = \theta^i_{\delta^i}]$, where $\mathbf{I}[\cdot]$ is the indicator function. In this latter case, there is no uncertainty, and π^i and σ^i are not used.

We set $\boldsymbol{\pi}$ to a fixed value that we learn by cross-validation. Thus, all $f_\theta(\theta^i|\delta^i = 0, \pi^i)$ are a Gaussian distribution with the same parameters. We could find the more likely $\boldsymbol{\pi}$ given the observations, but it is out of the scope of this paper. From a practical perspective, it suffices to assume that $\boldsymbol{\pi}$ takes a fixed value to achieve good performance in practice.

5. Selection of $\boldsymbol{\delta}$

In this section we describe the selection of $\boldsymbol{\delta}$. The algorithm starts from $\boldsymbol{\delta} = \mathbf{0}$, and it sequentially determines which potential to compute next, until the time budget, t_{total}, expires. We denote the known potentials at time t as $\boldsymbol{\theta}_{\boldsymbol{\delta}_t}$. The algorithm ranks the unknown potentials with a score, and thus prioritizes the potentials in the time budget. This is done by selecting the potentials with higher score. We summarize all steps in Alg. 1.

Let $S^i_{\boldsymbol{\delta}_t}$ be the score that ranks the potentials. We define $S^i_{\boldsymbol{\delta}_t}$ as the expected reward of instantiating the potential i. This is

$$S^i_{\boldsymbol{\delta}_t} = \mathbb{E}_\theta \left[R \left(P \left(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}_t} : \theta^i = \theta \right) \right) \right], \qquad (7)$$

where the expected value is over $\theta \sim f_\theta \left(\theta^i|\delta^i = 0, \pi^i\right)$, which is the Gaussian model of the posterior $P(\theta^i|\boldsymbol{\theta}_{\boldsymbol{\delta}_t})$. We use $\boldsymbol{\theta}_{\boldsymbol{\delta}_t} : \theta^i = \theta$ to indicate that θ^i in $\boldsymbol{\theta}_{\boldsymbol{\delta}_t}$ has been set to θ. $R(\cdot)$ is the reward of instantiating $\theta^i = \theta$, and it evaluates the probability distribution of $\mathbf{X}^\star$. The reward prioritizes probability distributions using a pre-defined criterion, such as having low uncertainty in the labeling of $\mathbf{X}^\star$. There are different possible criterions to define it, and we analyze two of them in the sequel. Observe that Eq. (7) evaluates the expected value of the reward by sampling θs from $f_\theta(\theta^i|\delta^i = 0, \pi^i)$, and evaluating the reward we would get if θ^i is clamped to the sampled θ.

Algorithm 1: Active MAP

```
$\boldsymbol{\delta}_0 = \mathbf{0}$;
while $t<t_{total}$ do
    ▷ Compute the score for the Unkown Unary Potentials:
    forall the $\delta^i_t = 0$ do
        $S^i_{\boldsymbol{\delta}_t} = \mathbb{E}_\theta \left[ R \left( f_{PM} \left( \mathbf{x}|\boldsymbol{\delta}_t : \delta^i = 1, \boldsymbol{\pi}_t : \mu^i = \theta \right) \right) \right]$
    end
    ▷ Instantiate the Unary Potential with higher $S^i_{\boldsymbol{\delta}_t}$:
    $i^\star = \arg\max_i S^i_{\boldsymbol{\delta}_t}$
    $\delta^{i^\star} = 1$, Compute $\theta^{i^\star}$
end
$\mathbf{x}^\star = \arg\max_{\mathbf{x}} f_{PM}(\mathbf{x}|\boldsymbol{\delta}, \boldsymbol{\mu})$
```

We can further develop $S^i_{\boldsymbol{\delta}_t}$ in Eq. (7). According to Prop. 1, $P\left(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}_t} : \theta^i = \theta\right)$ is a PM, which is $f_{PM}\left(\mathbf{x}|\boldsymbol{\delta}_t : \delta^i = 1, \boldsymbol{\pi}_t : \mu^i = \theta\right)$. Thus, $S^i_{\boldsymbol{\delta}_t}$ becomes

$$S^i_{\boldsymbol{\delta}_t} = \mathbb{E}_\theta \left[R \left(f_{PM} \left(\mathbf{x}|\boldsymbol{\delta}_t : \delta^i = 1, \boldsymbol{\pi}_t : \mu^i = \theta \right) \right) \right], \qquad (8)$$

Below, we introduce two possible criterions for the expected reward, and analyze the computational cost of calculating the reward.

5.1. Expected Reward

To compute the reward, we adapt two standard techniques from the active learning literature [18], namely the residual entropy and the labeling change. In the following we discuss them in the context of active MAP inference, and we show that both criteria can be effectively computed from a set of samples derived from the PM.

Expected Residual Entropy (ERE). We can compute the reward using the residual entropy in order to reduce the uncertainty of the MAP labeling. Then, the reward $R(\cdot)$ becomes $-H\left(f_{PM}\left(\mathbf{x}|\boldsymbol{\delta}_t : \delta^i = 1, \boldsymbol{\pi}_t : \mu^i = \theta\right)\right)$, where $H(\cdot)$ is the entropy, and can be computed by drawing samples from the PM. Note that reducing the uncertainty of the MAP labeling does not necessarily mean that the labeling is closer to the true MAP labeling.

Expected Labeling Change (ELC). [22] proposes to evaluate the expected change in the labeling. In the case of our problem, it is the change in the labeling induced from instantiating a potential. Thus, the reward $R(\cdot)$ is $\Delta\left(\mathbf{x}^\star_t, \arg\max_{\mathbf{x}} f_{PM}(\mathbf{x}|\boldsymbol{\delta}_t : \delta^i = 1, \boldsymbol{\pi}_t : \mu^i = \theta)\right)$, where $\mathbf{x}^\star_t$ is the MAP labeling at iteration t, and $\Delta(\cdot, \cdot)$ is a function that counts how many labels of $\mathbf{x}^\star_t$ differ from the labeling that we obtain with the PM when instantiating $\theta^i = \theta$.

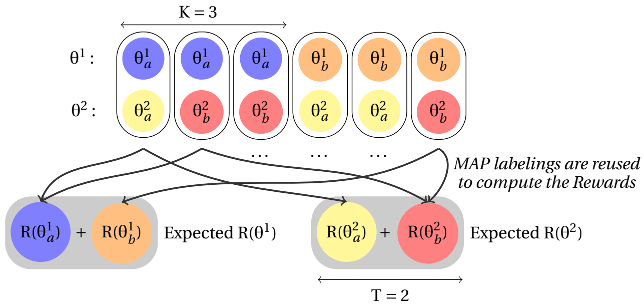


Figure 3. *Illustrative example of the TK samples of $\boldsymbol{\theta}$ to compute the expected Reward for all unknown potentials (best seen in color).* Example with 2 unknown potentials, $T = 2$ and $K = 3$.

5.2. Efficient Computation of the Reward

We can see by analyzing Eq. (8), that in the calculation of the expected reward, there are TK computations of MAP inference, where T is the number of samples of θ^i, and K the number of samples of the PM. This is because for the T samples of θ^i, we evaluate a PM that computes MAP K times. Thus, the cost of computing the scores for a number U of unknown potentials is $O(TKUm)$, where m is the cost of inferring the MAP labeling.

According to Alg. 1, the scores are evaluated every time we instantiate a potential. Thus, if doing TKU times MAP has a comparable cost to instantiate one potential, rather than speeding up the whole pipeline, Active MAP may become the computational bottleneck. In the following, we introduce two complementary strategies that render the evaluation of the scores efficient in practice.

Efficient computation of the expected reward. We first introduce a strategy to reduce $O(TKUm)$ to $O(TKm)$. It is based on the observation that the PM draws the unknown parameters of the energy from $f_{\boldsymbol{\theta}}\left(\theta^i|\delta^i = 0, \pi^i\right)$, which is the same probability distribution that we use to generate the θs for the expected value. Thus, we could reuse the same samples of $\boldsymbol{\theta}$ to calculate both the expected value, and the energy function of the different perturbations of the PM.

Recall that in the expected reward, for each θ^i, PM computes MAP inference K times with μ^i fixed to θ^i. We can generate a set of TK samples that can be used for all PMs of any unknown potential. This is feasible by drawing T values of θ^i from $f_{\boldsymbol{\theta}}$, and extending them, by repeating those T initial values of θ^i by K times, in a random order. We do this for all unknown potentials and obtain a set of TK different vectors of $\boldsymbol{\theta}$. Fig. 3 is an example of $\boldsymbol{\theta}$s generated in that way. Note that for each value of θ^i, we always have K different samples of $\boldsymbol{\theta}$, having the unknown potentials perturbed and θ^i fixed. This coincides with the form of the energy function of the K perturbations of the PM.

The limitation of this method is that θ^i takes only T different values in the TK samples, and they might not be diverse enough to correctly estimate the reward. Yet, we observed in the experiments that this approach achieves the same performance as using TKU different samples, even with small K and T.

Algorithm 2: Active MAP with Area of Influence

```
$\boldsymbol{\delta}_0 = \mathbf{0}$;
while $t < t_{total}$ do
    ▷ Compute the score for the Unkown Unary Potentials:
    forall the $\delta^i_t = 0$ do
        $S^i_{\boldsymbol{\delta}_t} = \mathbb{E}_\theta\left[R\left(f_{PM}\left(\mathbf{x}|\boldsymbol{\delta}_t : \delta^i = 1, \boldsymbol{\pi}_t : \mu^i = \theta\right)\right)\right]$
    end
    while $\exists S^i_{\boldsymbol{\delta}_t} \neq -\infty$ do
        ▷ Instantiate the Unary Potential with higher $S^i_{\boldsymbol{\delta}_t}$
        $i^\star = \arg\max_i S^i_{\boldsymbol{\delta}_t}$
        $\delta^{i^\star} = 1$, Compute $\theta^{i^\star}$
        ▷ Delete Candidates from the Area of Influence:
        forall the $x^j \in$ A. Infl. $(\delta^i = 1, \{\mathbf{x}\}_{K^2})$ do
            $S^i_{\boldsymbol{\delta}_t} = -\infty$
        end
    end
end
$\mathbf{x}^\star = \arg\max_{\mathbf{x}} f_{PM}(\mathbf{x}|\boldsymbol{\delta}, \boldsymbol{\mu})$
```

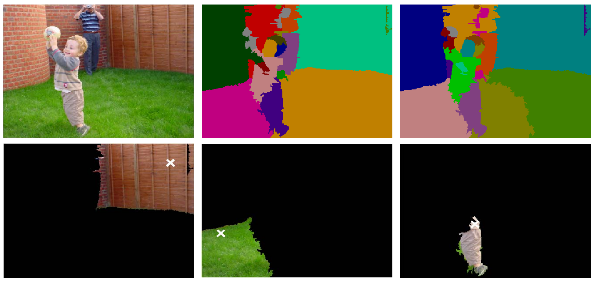
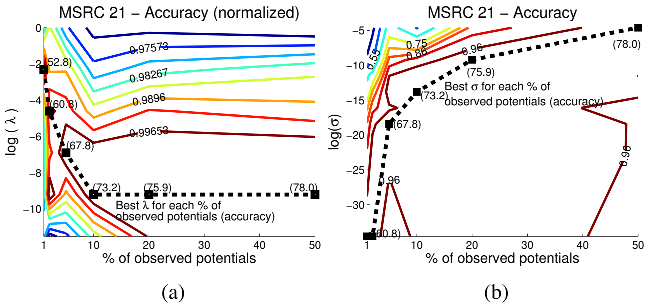


Figure 4. *Area of Influence and Expected Reward.* The top row shows the original image and two samples of PM with 0% of observed unary potentials. The bottom row shows the area of influence of the first selected superpixels that ranked higher with ELC.

Area of Influence. We propose a simple strategy to avoid re-computing the scores every time we instantiate a potential. It is summarized in Alg. 2. It is assumed that instantiating a potential reduces the score of the potentials that are in its "area of influence", while the rest remain unchanged. Under this assumption, only the scores in the area of influence are unreliable if they are not recomputed. We discard such scores as candidates until we re-compute the scores. This is done at the point that all potentials have been discarded.

We define an heuristic way, yet effective, to compute the area of influence. We define the area of influence as set of nodes that in all samples drawn from $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta}_{\boldsymbol{\delta}})$ take the same labeling value, and form a connected blob in the image. In Fig. 4 we illustrate several examples of the estimated area of influence for some potentials, and in the experiments section we show that using the area of influence is as effective as not using it, but it yields dramatic speed ups. Note that computing the area of influence does not incorporate any extra major cost, since we can use the samples of the PM used for calculating the rewards.



(a) (b)

Figure 5. *Learning the Parameters on MSRC-21.* (a) Impact of λ and (b) σ when varying the percentage of instantiated potentials. The accuracy is normalized by the maximum accuracy for each amount of observed potentials.

6. Experiments

We report results of our method on two popular datasets for semantic segmentation, namely the PASCAL VOC 2010 [5] and MSRC-21 dataset [19]. We use the standard evaluation set up. We first describe the implementation details and discuss the computational times (with a CPU 2.8GHz i7 with 8 cores). Then, we analyze the impact of the parameters and the heuristics we use, and we report results on the two datasets. Finally, we slightly modify the experimental setup and we show active MAP for human-in-the-loop semantic segmentation.

6.1. Implementation and Computational Time

We use a typical CRF with Potts pairwise potential modulated by the difference in color [6]. We use Active MAP to select which unary potentials to compute, since they have the higher computational load. The smoothness potentials are always computed, and thus, $\boldsymbol{\delta}$ is initialized to include the smoothness potentials. Below we describe each of the pipeline components for semantic image segmentation.

Unsupervised Segmentation. We first over-segment the images using SLIC superpixels [1], which allows us to work at superpixel level. The VOC10 images are over-segmented with about 800 superpixels, and for MSRC-21 we use about 300. SLIC takes on average 0.2 seconds per image.

Unary Potentials. In order to show that state-of-the-art methods can benefit from Active MAP, we use the publicly available features and classifiers in [12]. It extracts features taking into account the context of the image at different scales. Overlapping patches are described with SIFT and RGB histograms, and are encoded using Bag-of-Words (BoW) at 6 different contextual scales around the superpixels. The classifiers for each unary potential are SVMs with intersection kernel. In [12] they showed that with these features, they achieve comparable performance with or without using a CRF.

The computational cost for the different parts in VOC10 are 0.11s to compute dense SIFT, 0.05s to compute dense RGB histogram over the patches, and 0.6s to build all the BoW of an image with a fast nearest neighbor extraction. For MSRC-21 these costs are 0.03s to compute dense SIFT, 0.01s to compute RGB histograms, and 0.06s to build the BoWs. The cost of computing these features can not be saved by the Active MAP inference, because we use a global classifier that uses features over the entire image.

In the case of VOC10, computing the classification score with an SVM for a superpixel takes 0.02 seconds, and, hence, for an image with 800 superpixels this takes 16 seconds. In MSRC-21, computing the classification score for each superpixel takes 0.01 seconds, and for 300 superpixels 3 seconds. Most of the classifier costs - *i.e.* the main bottleneck of the pipeline - can be saved by Active MAP inference.

Smoothness Potentials. It is a Potts model modulated by the difference of the mean of the RGB color of the connected superpixels. It takes 0.1 seconds to compute for 800 superpixels, and 0.03 seconds for 300 superpixels. We denote the weight that multiplies the smoothness term in the energy function as λ, and it is one of the parameters that we learn in the following section.

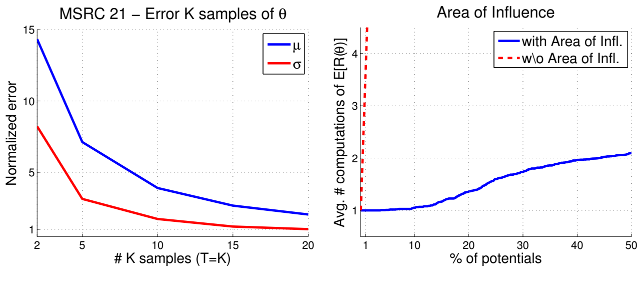
Inference. We use α-expansion graph cuts [3] to compute the MAP labeling in a complete energy function. For the PM we use $K = 5$ samples, which takes 0.03 seconds for VOC10 and 0.02 seconds for MSRC-21. For the expected reward we use $T = 5$, and it takes 0.15 and 0.12 seconds in total, respectively. The final labeling $\mathbf{x}^\star$ is computed with $T = 1$, *i.e.* a single time MAP inference.

6.2. Learning the Parameters

The parameters that we learn are the weight of the smoothness potentials (λ), and the model of the missing unary potentials ($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$). We learn them by cross-validation in the validation set, depending on the amount of instantiated potentials and the specific reward we use. In the following, we show the results in MSRC-21 when using the ELC reward. For VOC10 and the other rewards, we follow the same procedure.

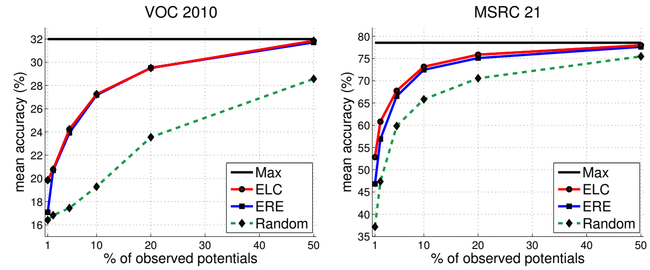
Weight of the Smoothness Potential (λ). In Fig. 5a we report the impact of λ on the accuracy. We can see that depending on the amount of instantiated potentials, the optimal value for λ may vary (indicated with the black line). Note that when few potentials are instantiated, the value of λ increases. This is because higher λ encourages label propagation, which is more important when we have less observations. When all potentials are observed, setting λ to 0 or very little gives the best performance, which is in accordance to [12]. We use the best λ for each amount of instantiated potentials.

Model of the Missing Parameters ($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$). We may use $\boldsymbol{\mu}$



(a) (b)

Figure 6. *Impact of the Heuristics on MSRC-21.* (a) Average squared deviation (error) from the mean and variance of the set of K samples used for a PM in the calculation of the reward for one unknown unary potential, when $T = K$. The error is normalized to the error of not using the heuristic. (b) Average amount of times the scores are calculated when varying the percentage of observed unary potentials.



(a) (b)

Figure 7. *Results on (a) VOC10 and (b) MSRC-21.* Accuracy when varying the percentage of instantiated potentials.

Method	Global	Average	Features	Inference	Total	Average	Features	Inference	Total
	MSRC-21: Test Set					VOC10: Validation Set			
All CRF	78	**78**	3s	0.02s	3.02s	**32.9**	16s	0.03	16.03s
All max	78	**78**	3s	—	3s	32.0	16s	—	16s
ELC 20%	76	**76**	0.6s	0.34s	0.94s	**29.5**	3.2s	0.37s	3.57s
ERE 20%	76	75	0.6s	0.34s	0.94s	29.5	3.2s	0.37s	3.57s
Random 20%	72	70	0.6s	0.1s	0.7s	23.5	3.2s	0.12s	3.32s
ELC 5%	70	**68**	0.15s	0.34s	0.49s	**24.2**	0.8s	0.12s	0.92s
ERE 5%	69	67	0.15s	0.34s	0.49s	23.9	0.8s	0.12s	0.92s
Random 5%	65	60	0.15s	0.1s	0.25s	17.4	0.8s	0.03s	0.83s
	MSRC-21: Human-in-the-loop					VOC10: Test Set			
All	98	**97**	—	—	300 clicks	33.5	16s	—	16s
ELC 20%	94	**92**	—	0.34s	60 clicks	30.4	3.2s	0.37s	3.57s
ELC 5%	86	84	—	0.34s	15 clicks	24.8	0.8s	0.12s	1.17s
ELC 1%	67	67	—	0.34s	3 clicks	—	—	—	—

Table 1. *Summary of all the results in MSCR-21 and VOC10.* The average score provides the per-class average. The time measurements are for one image.

to enforce a prior distribution over the classes. Yet, since the datasets we evaluate have only about 20 object classes, using this prior distribution can artificially boost the performance. Thus, we set all entries of $\boldsymbol{\mu}$ to the same constant value, which only adds an offset to the energy function that has no effect on the MAP labeling.

In Fig. 5b we show the impact of $\boldsymbol{\sigma}$ on the accuracy, when varying the percentage of observed unary potentials. $\boldsymbol{\sigma}$ is the level of injected noise in $P(\mathbf{X}^\star = \mathbf{x}|\boldsymbol{\theta_\delta})$, which is necessary to effectively evaluate the expected rewards. Note that when there are more potentials instantiated, the optimal $\boldsymbol{\sigma}$ increases. This might be to calibrate the amount of injected noise in the energy when less potentials can be perturbed.

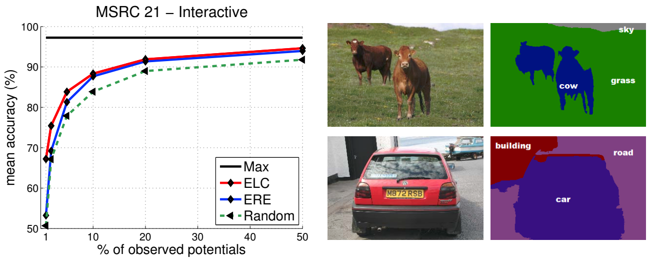
6.3. Results

Efficient Computation of the Expected Rewards. We analyze the impact of the heuristics we introduced in Sec. 5.2. In Fig. 6a we show the error of the mean and variance of the K samples of θ^i when reusing the samples generated for the expected value. Ideally, the samples of θ^i follow a Gaussian distribution, but due to the heuristic we use to generate them, the samples could have deviated from the original Gaussian distribution. We evaluate the average squared deviation (error) from the mean and variance of the set of K samples used in the calculation of the reward, for one unknown unary potential. The average is over all unknown potentials of all images, when there are no instantiated potentials. We set $T = K$ to proportionally increase the amount of samples. The error is normalized to the error of not using the heuristic. Note that as expected, the normalized error tends to 1 (same error as not using the heuristic) when we increase the amount of samples. In the experiments, we use $T = K = 5$ samples because it is a good tradeoff between computational cost and accuracy.

In Fig. 6b we analyze the impact of using the area of influence (Alg. 1 compared to Alg. 2). Recall that we discard the potentials that are in the area of influence of an instantiated potential, and we recompute the scores when all potentials have been discarded. We report the average number of times the scores are computed when varying the number of observed unary potentials. We can see that with 50% of the nodes instantiated, it only computes the scores 2 times (in average for all images). Note that this is a dramatic reduction of the computational cost, since without area of influence, the number of times it needs to compute the scores increases linearly to the number of observed unary potentials. Additionally, we observed that both methods obtain the same accuracy (we could only compare up to 10% of instantiated potentials due to the high computational cost of not using the area of influence).

Active MAP for semantic segmentation. We report results on MSRC-21 and VOC10, of the active MAP inference, with the ERE and ELC, and randomly selecting the unary potentials to compute the classifiers (referred as *Random*). We also report the results of using all the unary potentials and taking the maximum value of each, referred as *Max*, and when having the complete CRF, referred as *All CRF* in the tables.

In Fig. 7 we show the evolution of the performance when increasing the amount of instantiated potentials on MSRC-21 and VOC10 (on the validation set), and in Table 1 we report more detailed results on the MSRC-21 dataset and VOC10 (validation and test set). We also report the times for computing the features and classifiers related to the potentials, and the inference time which includes the overhead of computing the active MAP inference. We can see that on VOC10, the Active MAP with ELC reward, yields a speed-



(a) (b)

Figure 8. *Results on MSRC-21 with a human in the loop.* (a) Average accuracy, and (b) example of resulting images.

up of around 20x when only instantiating 5% of the unary potentials, achieving very competitive results. When instantiating only 20% of the unary potentials, there is a speed-up of 5x, and the performance only decreases about 3% with respect to computing all the unary potentials. Note that the overhead of extra computation of Active MAP is very small for all cases. The ELC achieves slightly better accuracy than ERE reward, specially with fewer observed unary potentials, and both methods outperform the Random strategy. In Fig. 1 we show images of the results achieved with Active MAP inference for different time budgets on VOC10.

Active MAP for human-in-the-loop segmentation. We evaluate the case of having the true labeling for some superpixels. This could be the case of having some unary potentials that may be prohibitive to compute, or also, when Active MAP interacts with a human that is asked the ground-truth for some superpixels. We slightly modify the set up used in previous experiments, by setting the instantiated unary potentials to add high penalties for the labels different from the ground-truth, or 0 otherwise.

In Fig. 8 and also in Table 1 we report results on MSRC-21. We include the case that all the unary potentials are known with the true label (refered as *Max*), which gives an upper-bound of the performance limited by the errors introduced by the superpixels. We can see that with 2% of the superpixels, which is about 6 superpixels in the image, we obtain the same performance as the state-of-the-art method [2].

7. Conclusions

We presented a method for active MAP inference on a CRF with unknown parameters. We showed its relation to the Perturb-and-MAP random field. The method incrementally adds the most promising parameters to the energy function using ranking criteria borrowed from active learning. Experiments on various datasets show that active MAP inference leads to significant computational savings, that clearly compensate for the overhead of computing the complete set of parameters of the energy function. The proposed method is useful when the computation of the energy function is more demanding than the MAP inference, as is often the case in semantic image segmentation. A research line that we are pursuing, and that we did not exploit in this paper, is to integrate dynamic inference techniques into our method.

Acknowledgments. This work was supported by the EU projects RADHAR (FP7-ICT-248873) and IURO (FP7-ICT-248314). We also thank ERC support from AdG VarCity. S. Ramos acknowledges the support of TRA2011-29454-C03-01. We also thank D. Tanprayoon and the anonymous reviewers for their valuable comments.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 2012.

[2] X. Boix, J. M. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez. Harmony potentials - fusing global and local scale for semantic image segmentation. *IJCV*, 2012.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004.

[4] G. Csurka and F. Perronnin. An efficient approach to semantic segmentation. *IJCV*, 2010.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010.

[6] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

[7] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 2008.

[8] J. Jiang and Z. Tu. Efficient scale space auto-context for image segmentation and labeling. In *CVPR*, 2009.

[9] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

[10] V. Kolmogorov and M. J. Wainwright. On optimality properties of tree-reweighted message-passing. In *UAI*, 2005.

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

[12] A. Lucchi, Y. Li, X. Boix, K. Smith, and P. Fua. Are spatial and global constraints really necessary for segmentation? In *ICCV*, 2011.

[13] M. Maire, S. X. Yu, and P. Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011.

[14] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.

[15] C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. In *ECCV*, 2008.

[16] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, 2011.

[17] N. Plath, M. Toussaint, and S. Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *ICML*, 2009.

[18] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648. University of WisconsinMadison, 2009.

[19] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009.

[20] D. Tarlow, R. Adams, and R. Zemel. Randomized optimum models for structured prediction. In *AISTATS*, 2012.

[21] J. Verbeek and B. Triggs. Scene segmentation with conditional random fields learned from partially labeled images. In *NIPS*, 2007.

[22] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *CVPR*, 2012.

[23] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory*, 2005.