On-line Semantic Perception Using Uncertainty

Roderick de Nijs[1,*], Sebastian Ramos[1,*], Gemma Roig[2], Xavier Boix[2], Luc Van Gool[2], Kolja Kühnlenz[1,3]

[1]Institute of Automatic Control Engineering
Technische Universität Munchen
80333 Munich, Germany
{rsdenijs,koku}@tum.de, jsramosp@mytum.de

[2]Computer Vision Laboratory
ETH Zürich
8092 Zurich, Switzerland
{gemmar,boxavier,vangool}@vision.ee.ethz.ch

Abstract— **Visual perception capabilities are still highly unreliable in unconstrained settings, and solutions might not be accurate in all regions of an image. Awareness of the uncertainty of perception is a fundamental requirement for proper high level decision making in a robotic system. Yet, the uncertainty measure is often sacrificed to account for dependencies between object/region classifiers. This is the case of Conditional Random Fields (CRFs), the success of which stems from their ability to infer the most likely world configuration, but they do not directly allow to estimate the uncertainty of the solution. In this paper, we consider the setting of assigning semantic labels to the pixels of an image sequence. Instead of using a CRF, we employ a Perturb-and-MAP Random Field, a recently introduced probabilistic model that allows performing fast approximate sampling from its probability density function. This allows to effectively compute the uncertainty of the solution, indicating the reliability of the most likely labeling in each region of the image. We report results on the CamVid dataset, a standard benchmark for semantic labeling of urban image sequences. In our experiments, we show the benefits of exploiting the uncertainty by putting more computational effort on the regions of the image that are less reliable, and use more efficient techniques for other regions, showing little decrease of performance.**

I. Introduction

Semantic labeling of a dynamic environment aims at assigning a semantic class to each region in an image. It is of crucial importance for the design of intelligent robots which are able to understand their surroundings.

The semantic labeling of different regions in an image usually makes use of contextual information, because local image patches suffer from perceptual aliasing [1]. Many recently proposed methods exploit contextual information by expressing the inter-dependencies of different regions in the scene through CRFs. These are probabilistically sound models that have been successfully used in numerous applications, and have become the *de facto* model for incorporating spatial or conceptual dependencies between variables. The reason for the popularity of CRFs is that they allow the designer to encode conditional independence assumptions and are highly flexible in terms of the features that can be incorporated in the model.

The process of inferring the most likely labeling in a CRF, denoted as the Maximum-A-Posteriori (MAP), can be computed very efficiently, and many methods have been developed for minimizing the so-called energy function of the problem [2], [3]. However, looking only at the MAP solution, one neglects the probabilistic properties of the model and reduces it to a pure optimization problem, thereby ignoring the reliability of that particular solution in comparison to others.

Neglecting the probabilistic form of the solution is undesirable for real world-embedded robots. In such systems, interconnected modules often combine information from multiple sources, and the uncertainty is delivered to decision making or planning modules that decide on the next action to take. The initial processing stages usually use the uncertainties for information fusion, and high level modules use it for making more sensible or safe decisions. Furthermore, a robot might benefit from knowing the uncertainty of the semantic labeling by using it as indicator of where to focus the computational or attentional effort.

There are various alternatives that can be used to estimate the uncertainty of the labeling. For instance, one possibility is directly taking the scores of individual semantic classifiers, without the CRF. A higher score means a higher confidence about the object being detected, allowing to associate a confidence value to the detection. However, the individual semantic classifiers are not able to encode the structure of the labeling, which is necessary to encode contextual dependencies. Another way is to use methods that estimate the CRF density function, e.g., max-sum loopy belief propagation, which is able to approximate the marginals [4]. Also, Markov Chain Monte Carlo (MCMC) sampling could be used, which in theory is able to deliver any distribution of interest. However, these techniques are often avoided because belief propagation on loopy, densely connected, graphs has no guarantee on convergence nor quality of the results, and MCMC sampling in a CRF needs many iterations to guarantee a good mixing of the chain. Remarkably, Kohli and Torr [5] introduced a method to estimate the uncertainty of the max-marginals with graph-cuts, but these do not correspond to the marginals, and it is not suitable for temporal smoothness [6].

In this paper, we introduce a model for semantic segmentation of a dynamic image sequence, and show how we

[*]Both first authors contributed equally.

[3]Institute for Advanced Study, Technische Universität München, 80333 Munich, Germany

This work was partly supported by the EU projects FP7-ICT-24314 Interactive Urban Robot, and FP7-ICT-248873 Robotic ADaptation to Humans Adapting to Robots, by the DFG excellence initiative research cluster Cognition for Technical Systems CoTeSys, and by the Institute for Advanced Study (IAS), Technische Universität München.

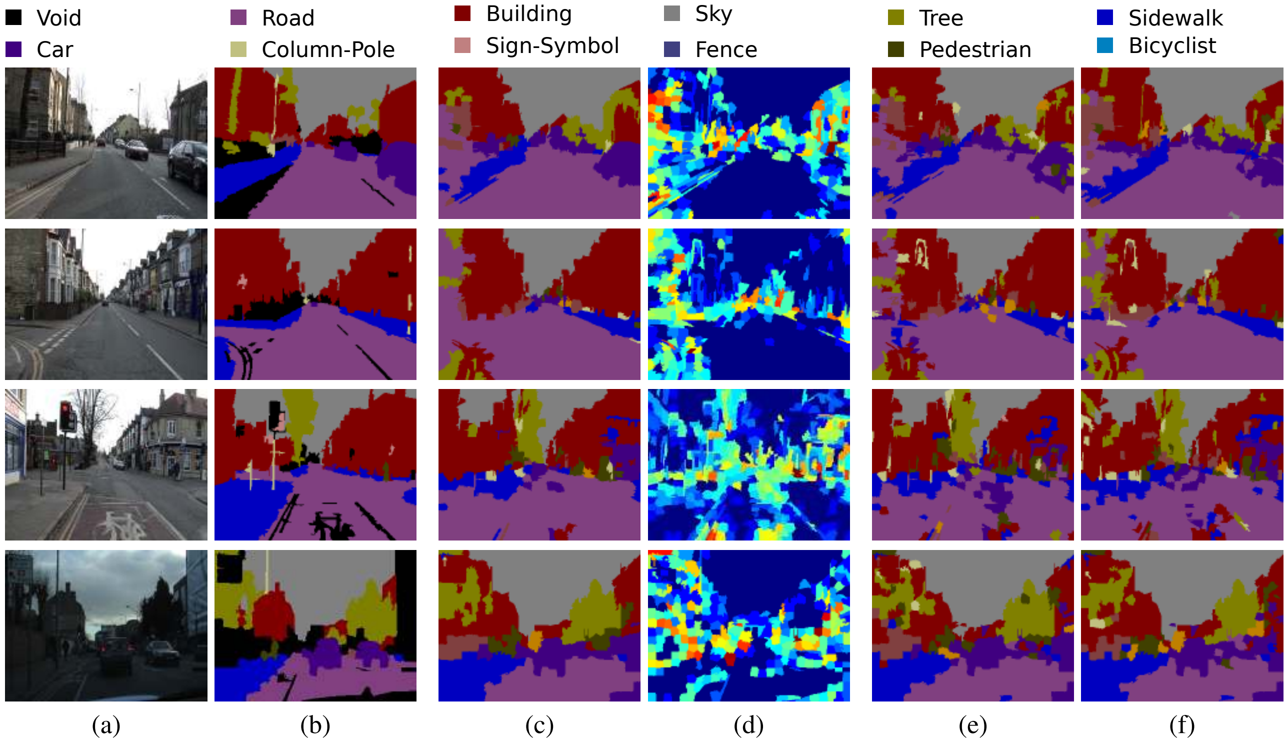

Fig. 1. Example of a labeled image of a dynamic sequence with an on-line semantic labeling framework. In column (a) there is the original image, in (b) the ground truth labels, (c) is the labeling solution of the MAP, (d) its measure of uncertainty with the entropy of the marginals computed from the samples obtained with the PM, in which reddish regions indicates more uncertainty. Finally, (e) and (f) are two samples drawn with the PM.

can compute the confidence of the labels from the marginal distribution obtained from a Perturb-and-Map Random Field model (PM) [7]. This recently presented probabilistic model allows to draw samples efficiently by introducing perturbations in the energy function and obtaining the corresponding MAP estimate. Also, we exploit the uncertainty of the labeling by concentrating the computational effort in the most uncertain regions of the scene. This speeds up the system by releasing computational effort from confident regions with a minimal sacrifice in accuracy. In Fig. 1 we illustrate examples of several ground truth labels, the MAP solution, labeling examples of an image obtained from the PM and the entropy computed based on the samples.

We tested our method in a standard benchmark of semantic video labeling, namely the CamVid dataset [8]. We show that drawing ten samples of the model is enough for making a good prediction on the confidence level of individual image regions. We also show that there is only little loss of performance when using powerful and computationally costly classifiers only on the locations with more uncertainty, and weaker and faster classifiers on the rest of the image regions.

II. Related Work on Semantic Video Labeling

Semantic labeling of image streams is an important problem for robotics and computer vision. Related to our approach are the fully supervised methods for semantic labeling in a temporal framework. Budvytis *et al.* [6] introduced a method to estimate the uncertainty in the labeling by computing the marginals in a probabilistic model. Nonetheless, this method is not for on-line applications, and the computation of the uncertainty is difficult to adapt to other, more general, models. Also, [9] showed promising results in semantic segmentation in videos, but it is neither on-line nor estimates the uncertainty. Finally, [10] is able to exploit the temporal redundancy to save computational time, but it does not rely on a temporal probabilistic model, and makes hard decisions at each frame. In contrast to previous approaches, our method is an on-line probabilistic model that is able to estimate and exploit the uncertainties.

Another important strand of research focuses on the unsupervised labeling of a video. It differs from our method since it does not learn the semantic classes of the labels, and most of these unsupervised methods assume that all frames are available at processing time [11], [12], which does not hold in a robot, or only consider two frames at the same time [13], [14]. Vazquez-Reina *et al.* [15] introduced an unsupervised on-line method that takes into account a buffer of the last images to infer the labeling in the current frame. We bring this idea to the problem of supervised semantic labeling and uncertainty extraction. Other methods ask a user for part of the ground-truth [16], [17], and propagate it to the unlabeled regions. Unfortunately, we can not use these methods for on-line semantic labeling in a robot.

III. Uncertainty in the MAP Labeling

In this section we introduce the model used for the semantic labeling and estimating the uncertainty, which is a Perturb-and-MAP Random Field. In the following subsections, we first introduce CRFs, and show that it is difficult to measure the uncertainty of the inferred labeling. Then, we revisit Perturb-and-MAP Random Fields (PM), and introduce how this model enables us to draw samples, and evaluate the uncertainty of the inferred labeling.

A. Conditional Random Fields

Many labeling algorithms have been modeled with a CRF because of its freedom in the design for modeling the interactions between random variables, while still allowing to perform MAP inference efficiently. Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be the graph that represents our CRF, where $\mathscr{V}$ is used for indexing the nodes that correspond to random variables, and $\mathscr{E}$ is the set of undirected edges representing compatibility relationships between random variables. A clique is a subgraph in which every node is connected to all other nodes in the subgraph. $\mathscr{C}$ is the set of cliques of the graph that are not a subset of any other clique, also known as maximal cliques.

We use $\mathbf{X} = \{X_i\}$ to denote the set of random variables, which has a cardinality of N. Each random variable takes a discrete value from a set of labels $\mathscr{L}$, and $\mathbf{x} \in \mathscr{L}^N$ is a possible state or instantiation of $\mathbf{X}$. $P_\theta(\mathbf{x})$ denotes the probability density distribution of a labeling modeled with the graph $\mathscr{G}$, where θ are the parameters of the distribution. According to the Hammersley-Clifford theorem, the probability density that satisfies the Markov properties with respect to the graph $\mathscr{G}$ is a Gibbs distribution. Thus, $P_\theta(\mathbf{x})$ is the normalized negative exponential of an energy function, i.e.,

$$P_\theta(\mathbf{x}) = \frac{1}{Z} \exp\left(-E\left(\mathbf{x}\right)\right), \tag{1}$$

where Z is the normalization, also called partition function. Observe that the value of the partition function, i.e., Z, is obtained by summing over all possible states of $\mathbf{X}$, which is not possible to compute in practice. Thus, we do not have direct access to the probability density function, though we know which states are more probable than others.

The energy takes the following form:

$$E(\mathbf{x}) = \sum_{c \in \mathscr{C}} \psi_c(\mathbf{x}_c; \theta), \tag{2}$$

where ψ_c are the potential functions of the maximal cliques $c \in \mathscr{C}$ and θ the parameters of the model. The energy function can be also expressed as a product of parameters θ and the sufficient statistics of the model, denoted as the mapping $\phi(\mathbf{x})$. Thus, the energy can be equivalently written as

$$E(\mathbf{x}) = \sum_{c \in \mathscr{C}} \psi_c(\mathbf{x}_c; \theta) = \theta^T \phi(\mathbf{x}), \tag{3}$$

where T denotes vector transpose. $\phi(\mathbf{x})$ is a reparametrization of $\{\psi_c\}$, such that the energy is a product between $\theta \in \mathbb{R}^M$ and $\phi(\mathbf{x}) \in \mathbb{R}^M$. This is a common transformation in the literature, and in the rest of the paper we use both notations indistinctively.

Let $\mathbf{x}^\star$ be the state that minimizes the energy function, or equivalently, the Maximum-a-Posteriori (MAP) labeling:

$$\mathbf{x}^\star = \arg \min_{\mathbf{x} \in \mathscr{L}^N} \theta^T \phi(\mathbf{x}). \tag{4}$$

For many useful energy functions in vision and robotics there are efficient solvers that optimize this energy function, or can give some guarantees about the distance to the minimum. Indeed, one of the reasons of the success of CRFs is the availability of off-the-shelf solvers that allows the design of sophisticated energy functions without the need of rethinking the optimization.

To evaluate the uncertainty of the MAP labeling we need to explore the probability function that models the labeling, for instance, by drawing samples from the probability density. However, in a CRF we do not have access to the probability density, even though we can compute its maximum. This is because the probability density of a CRF is normalized with the partition function, i.e., Z, which can not be computed in practice. We might draw samples from a CRF with a Markov Chain Monte Carlo (MCMC), but it suffers from the curse of dimensionality in practical problems. In the next subsection, we revisit Perturb-and-MAP Random Fields, which was introduced to alleviate the difficulties in sampling from a CRF distribution.

B. Perturb-and-MAP Random Fields

Possibly because of the above mentioned limitations, generating samples from a CRF has been rarely explored. Recently, however, Papandreou and Yuille introduced the PM random field [7], which was specifically designed to allow drawing samples, bypassing the expensive MCMC. In a follow up paper, Tarlow *et al.* [18] extended this idea to models more general than CRFs. In the following we briefly review the PM random field, but for a complete explanation we refer the reader to [7].

Perturb-and-MAP is built around the effective MAP inference algorithms commonly used in CRFs. It draws approximate samples by calculating the MAP of an energy function constructed by perturbing its parameters. We define the random variable ε, which takes values from $\mathbb{R}^M$, and it is used to *perturb* the parameters such that $\tilde{\theta} = \theta + \varepsilon$. MAP inference can be done after perturbing the energy function by simply computing $\arg\min_{\mathbf{x}} (\theta + \varepsilon)^T \phi(\mathbf{x})$.

Let $\mathscr{P}_{\mathbf{x}} - \theta$ be the set of perturbations ε that yield an energy function with minimum labeling equal to $\mathbf{x}$, i.e.,

$$\mathscr{P}_{\mathbf{x}} - \theta = \left\{ \varepsilon \in \mathbb{R}^M | \mathbf{x} = \arg \min_{\mathbf{x}' \in \mathscr{L}^N} (\theta + \varepsilon)^T \phi(\mathbf{x}') \right\}. \tag{5}$$

Then, the PM model defines the probability density of a labeling $\mathbf{x}$ as

$$f_{PM}(\mathbf{x}; \theta) = \int_{\mathscr{P}_{\mathbf{x}} - \theta} f_\varepsilon(\varepsilon) d\varepsilon, \tag{6}$$

where $f_\varepsilon(\varepsilon)$ is the probability density distribution of ε, and might be independent from the parameters θ. Observe that $f_\varepsilon(\varepsilon)$ inserts randomness in the parameters, and thus determines which energy function is optimized. Intuitively, the PM model assigns more probability to the labelings that are inferred for more likely perturbations. We can draw samples from the PM model by simply generating a perturbation and doing MAP inference. Thus, even though

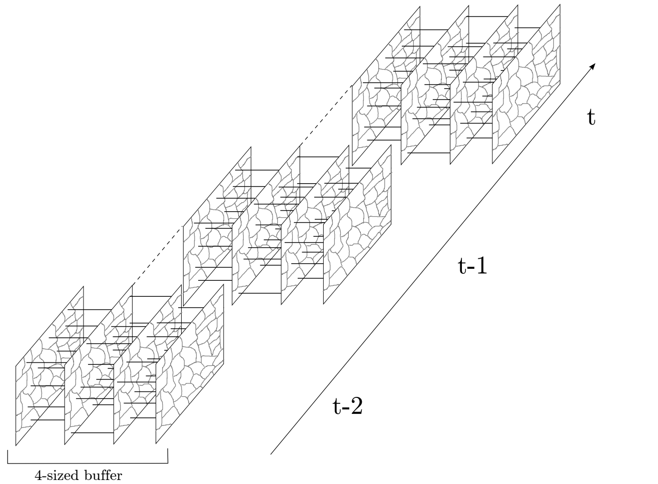

Fig. 2. Illustration of the on-line framework for semantic labeling of temporal sequences. We take a buffer of several frames to infer the labeling of the last taken frame. We use superpixel level instead of pixel level to define the random variables, and we take into account spatial smoothing and temporal consistency.

the exact computation of Eq. (6) is intractable in practice, we can efficiently generate samples.

C. Uncertainty of the MAP Labeling

We measure the uncertainty from multiple samples drawn from the PM model. Let $\{\varepsilon_j\}$ be the set of perturbations drawn from $f_\varepsilon(\varepsilon)$, and $\mathbf{x}(\varepsilon_j)$ the labeling obtained from one of those perturbations, i.e.,

$$\mathbf{x}(\varepsilon_j) = \arg\min_{\mathbf{x}\in\mathscr{L}^N} (\theta+\varepsilon_j)^T\phi(\mathbf{x}). \tag{7}$$

Thus, $\{\mathbf{x}(\varepsilon_j)\}$ is a collection of labelings that represents the density distribution of the PM model.

From the collection of samples we can extract some measurement of uncertainty. An interesting measure, that we will use in the paper, is the entropy of the marginal distributions, which gives an idea of the reliability of the labeling of each random variable. We estimate the marginal distribution by approximating it with a histogram. $Q_i(l)$ is the entry of the histogram of label $l \in \mathscr{L}$, that approximates the marginal of the random variable x_i:

$$Q_i(l) = \frac{1}{K}\sum_{\{\varepsilon_j\}} \mathbf{I}[x_i(\varepsilon_j) = l], \tag{8}$$

where $\mathbf{I}[\cdot]$ is the indicator function, and K normalizes the histogram. Then, the uncertainty can be computed with the entropy of the marginal, which is by definition: $H_i = -\sum_{l\in\mathscr{L}} Q_i(l)\log_2 Q_i(l)$.

IV. Semantic Labeling of a Dynamic Environment

In this section we introduce the formulation for the on-line semantic labeling problem. We aim at on-line labeling the video stream from a camera mounted on a robot navigating in a dynamic environment. Our formulation uses the Perturb-and-MAP Random Field to model the spatial and temporal consistency between and within frames of the video.

Instead of using one random variable for each pixel, a random variable is defined for each superpixel[1]. This strategy has been previously used to reduce the number of random variables of the model and speeds up inference, e.g., [19].

For each incoming frame we infer the best labeling by taking into account the previous F frames of the sequence. This allows temporal consistency in the last frames. In Fig. 2 we show a scheme of the on-line system with the superpixels.

A label is used to represent each semantic class, $\mathscr{L} = \{l_1, l_2 \ldots\}$. Each random variable is indexed using two indices, one for the set of frames $\mathscr{T}$ and another for the set of superpixels $\mathscr{P}$, which gives rise to $\mathscr{V} = \mathscr{T} \times \mathscr{P}$. In our approach, since we consider the last F incoming frames, it yields a set of frames as $\mathscr{T} = \{t_1, \ldots, t_F\}$. For a frame $t \in \mathscr{T}$, we denote as x_i^t a random variable associated with the superpixel i, $i \in \mathscr{P}$. We define $\mathscr{N}_i^t$ as the set of spatial neighbors of the random variable x_i^t which are in the same frame $t \in \mathscr{T}$, and $\mathscr{M}_i^t$ as the set of neighbors of x_i^t that are in the previous frame, i.e., in the frame $t-1$.

The energy function $E(\mathbf{x})$ is defined as the sum of the unary, the spatial smoothness and the temporal consistency potentials:

$$E(\mathbf{x}) = \sum_{t\in\mathscr{T}}\sum_{i\in\mathscr{P}} \psi_u(x_i^t) +$$
$$\sum_{t\in\mathscr{T}}\sum_{i\in\mathscr{P}}\sum_{j\in\mathscr{N}_i^t} \psi_s(x_i^t, x_j^t) + \sum_{i\in\mathscr{P}}\sum_{t\in\mathscr{T}}\sum_{j\in\mathscr{M}_i^t} \psi_{tc}(x_i^t, x_j^{t-1}), \tag{9}$$

In Fig. 3 we show the representation of the graph of our model with the notation used in this paper. The unary term $\psi_u(x_i^t)$ encodes the classification scores for superpixel i of frame t. The smoothness term $\psi_s(x_i^t, x_j^t)$ determines the pairwise relationship between neighboring nodes in $\mathscr{N}_i^t$. It represents a penalization for the labeling of two neighboring nodes. Finally, the temporal consistency potential $\psi_{tc}(x_i^t, x_j^{t-1})$ expresses the dependency relationship between the neighboring nodes $\mathscr{M}_i^t$ that are in consecutive frames. It enforces an agreement of the labeling among connected nodes between frames. In the following we explain the potentials of the energy function $E(\mathbf{x})$. For the implementation details we refer the reader to Section VI.

A. Unary Potentials

The unary potentials $\psi_u(x_i^t)$ are the scores obtained from the semantic classifiers applied at each superpixel independently. We use descriptors and classifiers based on [20], which are representative of the state-of-the-art in image semantic segmentation.

B. Spatial Smoothness Potentials

The smoothness term is a modulated Potts potential. We use the color differences between neighboring nodes to weight the potential. The potential penalizes neighboring

[1] A superpixel is a group of pixels that belong to the same object. A superpixel algorithm aims at over-segmenting the image by grouping pixels into superpixels.

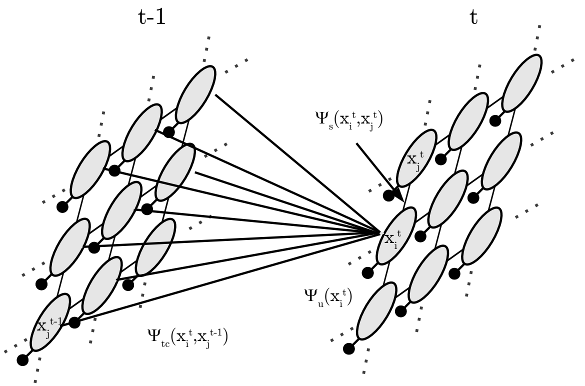

Fig. 3. Graph representation of our model with the notation used in this paper. For each superpixel i of frame t we define a random variable x_i^t. Each random variable has a unary potential $\psi_u(x_i^t)$ associated. $\psi_s(x_i^t,x_j^t)$ is the spatial smoothness potential for the neighboring nodes of the same frame. $\psi_{tc}(x_i^t,x_j^{t-1})$ is the time consistency term, which encourages an agreement between similar superpixels of consecutive frames.

nodes of the same image frame with different labels, depending on the color differences, i.e.,

$$\psi_s(x_i^t,x_j^t) = c_{ij}\mathbf{I}[x_i^t \neq x_j^t], \tag{10}$$

where $\mathbf{I}[\cdot]$ is an indicator function. c_{ij} is a similarity measure of the color between the superpixels indexed by i and j.

C. Temporal Consistency Potentials

Analogously to the spatial smoothness potential, we use a modulated Potts potential for the temporal consistency potential:

$$\psi_{tc}(x_i^t,x_j^{t-1}) = m_{ij}\mathbf{I}[x_i^t \neq x_j^{t-1}], \tag{11}$$

where m_{ij} is a similarity measure between superpixel i of frame t and superpixel j of frame $t-1$.

V. Exploiting Uncertainty for Efficiency

In this Section we show a simple application to exploit the uncertainty, which allows to speed up the process of labeling an image sequence of a dynamic environment. We aim at spending more computational effort in the areas where the semantic class is not so clear. However, applications of the uncertainty for other purposes are also feasible. For instance, a robot may make its exploration decisions based on the uncertainty coming from the vision system or actively employ other sensors to analyze uncertain regions.

We introduce a simple method, yet effective, aiming to strike a balance between efficiency and performance. When a new frame arrives, we set all unary potentials to values estimated from fast classifiers and image descriptors, and we compute the uncertainty. This gives an idea of which labels can already be inferred with the temporal and spatial redundancy, and which ones require more accurate unary potentials. We select which nodes will be assigned the accurate unaries by sampling without replacement from a categorical distribution, where each category is associated with a node and has a probability p_i, $i \in \mathscr{P}$, where $p_i = \frac{H_i^b}{T}$, $T = \sum H_i^b$ is a normalization factor and the exponent b is a parameter to control how much the samples should concentrate on uncertain nodes (b is set to 3 in the experiments). Once a pre-defined percentage of unaries is computed, we infer the MAP labeling. Computing a higher percentage of *accurate* unaries improves the inferred labeling, but at the cost of more computational effort.

VI. Implementation Details

In this section we describe the implementation details of the on-line semantic labeling system.

A. Unary Potentials

We first over-segment the images using the SLIC algorithm, which extracts superpixels [21]. The pipeline to calculate the unary potentials is shown in Fig. 4. First, we extract patches in a dense grid, independently of the superpixel form, and we describe and encode them. Then, for each superpixel, we pool the patches from the image region of the superpixel, and generate the feature descriptor of the superpixel. Finally, a classifier computes the score that is used as unary potential.

We define two different types of encodings, which are applied in different superpixels, depending on the entropy of the labeling (see Section V). One achieves state-of-the-art performance, but it is too computationally demanding (we refer to it as *accurate* feature). The second sacrifices some performance, but is more suitable for on-line applications (referred as *fast* feature). In the following subsections, we describe the details of each part of the pipeline for both features.

We extract patches in a dense grid in an image. The center of the patches are separated by 4 pixels, and are extracted at several scales ($8, 16, 32$ pixels of patch size). We describe each patch with a SIFT [22] descriptor, which has a length of 128, and it is used for both *fast* and *accurate* features.

The encoding is based on the Bag-of-Words approach. For the encoding of the *accurate* feature, we use the encoding based on [20]. For each input patch described with SIFT, we select the K nearest neighbors in a codebook, and we build a vector indicating the selected entries of the codebook. In the experiments, K is set to 5. The codebook is built by randomly picking a set of 1000 patches as codebook entries, which achieves performance similar to k-means clustering [23].

Feature encoding has been reported in the literature as an important bottleneck of the pipeline. A way to speed it up is using Nested Sparse Quantization for coding [24]. This method builds upon two sparse quantizations, which allows a very efficient encoding because it is based on binary vectors. This method is used for the *fast* feature. We use the parameters reported in [24].

The description of a superpixel is done by pooling the encoded SIFT features. We use two pooling regions, one inside the superpixel, and another of the contextual area around it. This contextual area is extended up to 4 times the size of the superpixel. This combination allows to include the context, which makes the descriptor more powerful, as shown in [19] and [20]. It gives a final descriptor of the superpixel of dimension $2 \times 1,000$ bins, due to the 2 pooling regions. We use use average pooling for the *accurate* feature,

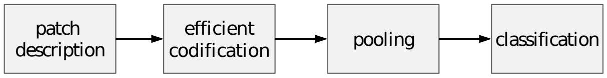

Fig. 4. Pipeline used for feature description and classification of a superpixel.

and max-pooling (cf. [25]) for the *fast* feature. In the case of semantic segmentation, we observed that max-pooling decreases the performance compared to average pooling, but the combination of Nested Sparse Quantization and max-pooling results in a binary vector, and performs well with linear (i.e., efficient) classifiers.

We obtain the classification scores for each superpixel with Support Vector Machines (SVM). For the *accurate* feature, we use the RB-χ^2 kernel, approximating its non-linear mapping using [26] for efficiency. For the *fast* feature we use a linear SVM. We learn the SVM using a one-vs-all scheme by using $5,000$ samples of the specific class as positive examples, and $5,000$ samples of the rest of the classes as negatives examples. The number of positive examples is less in case no sufficient training examples are available for a specific class. The parameter C of the SVM is set to $1,000$.

One of the main drawbacks of learning independent classifiers for multi-class problems is that at some point we have to merge the classification scores. Since each classifier is trained independently, the bias between classes is not taken into account . This effect is more noticeable when the training set is unbalanced. To alleviate this fact, we use a weight for the scores of each class to calibrate the confidence in the output of each classifier. In the learning stage, we perform a random search of the weights that maximize average accuracy.

B. Spatial Smoothness Potentials

To compute the color similarity between neighboring superpixels of the same frame, we use the norm of the difference between the mean RGB colors of the superpixels indexed by i and j. This is, $c_{ij} = \|c_i - c_j\|_2^2$, where c_i is the 3-dimensional vector of the mean of the RGB color of superpixel i.

C. Temporal Consistency Potentials

The matching between superpixels of consecutive frames is done based on the pairwise distance of the patches inside them. Thus, we compare the patches inside one superpixel with all patches inside the other superpixel, and we average the similarities to get a measure of the superpixel similarity. We use the BRIEF descriptor [27] of length 256 to compute the similarity measure. It uses binary descriptors that allows computing similarities with a Hamming distance, which are very efficient to evaluate, since most modern CPUs include dedicated hardware instructions [27].

D. Parameters

We use the Gumbel distribution for perturbing the energy function as suggested in [7] and perturb only the parameters associated with the unary potentials. The amount of samples are analyzed in the experiments section. The MAP inference is done with $\alpha-$expansion graph cuts [3]. In all experiments, the buffer of the on-line model takes the last 2 frames, because we did not observe significant improvements when including more frames.

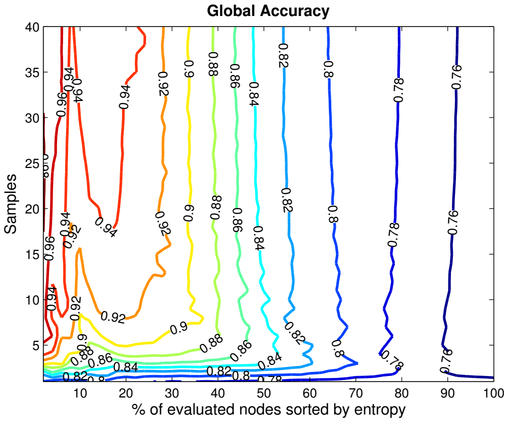

Fig. 5. Global accuracy rate on CamVid changing the number of samples of the perturb-and-MAP to compute the entropy of the marginals. We report results evaluating only the percentage of superpixels with less uncertainty up to a threshold.

The parameters to learn are the per-class weighting of the unary potentials, and the weights for the spatial smoothness and time consistency potentials. Parameters are learnt by performing a stochastic gradient descent step after observing each training example. The learning of the parameters of the CRF is done on the whole training set, with the unary potentials learned with 3-fold cross-validation.

VII. Experiments

We report results of our method on a challenging dataset for semantic segmentation in videos, namely the Cambridge-driving Labeled Video Database (CamVid) dataset [8]. It consists of a collection of 4 video sequences with ground truth, provided at a rate of 1Hz, and partially at 15Hz, of 32 semantic classes for each pixel of the frames. It has a total of 701 labeled images. We use the sequences 0016E5, 0006R0 and the first half of the dusk sequence (0001TP) for training, which has 468 images, and the sequence 0005VD and the second half of the dusk sequence for testing with 233 images.

We use the standard metrics to evaluate the accuracy performance of the different methods, namely the global and the per-class average accuracy rate. The global accuracy rate measures the percentage of correct classified pixels of all classes. The per-class average accuracy rate, takes into account the accuracy of the pixels of each of the class independently, and then it averages them.

A. Quality of the Uncertainty Measure

We evaluate the global accuracy metrics of CamVid only on the superpixels with less entropy than a threshold, sweeping the threshold such that it takes a percentage of nodes from 0% to 100%. In this way, we can evaluate the quality of the uncertainty measure. This experiment is performed with a varying number of samples ranging from 2 to 40. We measure the global accuracy metric with the entropy of the marginals explained in Section III. The results are reported in

	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Column	Sidewalk	Bicyclist	**Average**	**Global**
100%	59	75	93	84	45	90	53	27	0	55	21	54.7	75.0
0%	78	71	91	63	8	91	22	15	0	39	6	44.0	76.4
Entropy-based sampling													
30%	76	74	92	72	16	92	33	17	0	40	14	48.0	77.4
20%	77	73	91	70	14	92	33	18	0	40	12	47.3	77.2
15%	77	73	91	68	12	92	30	17	0	40	10	46.5	77.3
10%	77	72	91	67	11	92	28	16	0	40	8	45.7	77.0
5%	78	72	91	65	10	91	25	15	0	40	7	44.9	76.8
Random-based sampling													
30%	75	73	91	69	13	91	26	17	0	41	10	46.2	76.7
20%	76	72	91	67	11	91	27	17	0	40	9	45.7	76.6
15%	77	71	91	66	11	91	26	15	0	40	8	45.2	76.6
10%	77	71	91	65	9	91	25	15	0	40	7	44.7	76.5
5%	77	71	91	64	8	91	21	15	0	40	7	44.2	76.5

TABLE I

GLOBAL AND AVERAGE ACCURACIES ON THE TEST SET USING A PERCENTAGE OF SUPERPIXELS WITH THE ACCURATE FEATURES.

Fig. 5, showing the global accuracy rate as a function of the percentage of nodes that are considered for the evaluation and the amount of samples. First, we observe that the accuracy, when considering only nodes with low entropy, is very high. As more entropic nodes are considered, the global measure decreases until it reaches the global accuracy value of the whole test set. Furthermore, it can be seen that as more samples are considered, entropy becomes a more reliable indicator. Based on these observations we decide to use 10 samples for the rest of the experiments, giving a good compromise between accuracy and efficiency.

B. Exploiting the Uncertainty to Reduce the Computational Cost of the Labeling

In this experiment, we evaluate the method introduced in Section V. We use the uncertainty to choose where to compute the *accurate* features, whereas we use the *fast* features by default. In Table I, we compare the accuracy obtained choosing the nodes with the entropy versus random. This experiment shows that the uncertainty given by the entropy can be used to effectively estimate where to spend more computational time. In the table, 100% indicates that we only use the *accurate* features, achieving similar accuracy to the appearance-based state-of-the-art method for efficient applications [8], and 0% for the *fast* features. We see that thanks to the combination of the multiple potentials that bind the random variables together, the classification accuracy stays at reasonable levels with much less computational effort, and that our entropy based sampling outperforms random sampling in all situations. In terms of computation speed, using only the *fast* feature set we achieve 3 fps, while the accurate features achieve 0.2 fps, on an eight core machine, enabling online applications.

VIII. CONCLUSIONS

A model for on-line semantic labeling based on Perturb-and-Map was presented. This model is able to obtain a labeling, as well as to evaluate the uncertainty by computing the entropy of the marginals. Experiments demonstrated that this uncertainty measure correlates well with the accuracy of the classifier. It was shown in the experiments on the CamVid dataset that focusing computational effort in areas of high entropy compares favorably to a baseline. Thus, it pays to use powerful and costly features and classifiers on the locations with more uncertainty, and weaker and faster features and classifiers for the rest of the image regions. An interesting direction for future work might be to include higher-order cliques in the model, covering larger image regions in order to better predict which areas might require more attention.

REFERENCES

[1] S. Whitehead and D. Ballard, "Learning to perceive and act by trial and error," *ML*, 1991.

[2] B. J. Frey and D. J. C. MacKay, "A revolution: Belief propagation in graphs with cycles," in *NIPS*, 1997.

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

[4] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *IJCAI*, 2001.

[5] P. Kohli and P. H. S. Torr, "Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts," in *ECCV*, 2006.

[6] I. Budvytis, V. Badrinarayanan, and R. Cipolla, "Semi-supervised video segmentation using tree structured graphical models," in *CVPR*, 2011.

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

[8] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, 2008.

[9] P. Sturgess, K. Alahari, L. Ladicky, and P. H.S.Torr, "Combining appearance and structure from motion features for road scene understanding," in *BMVC*, 2009.

[10] J. Rituerto, A. Murillo, and J. Kosecka, "Label propagation in videos indoors with an incremental non-parametric model update," in *IROS*, 2011.

[11] W. Brendel and S. Todorovic., "Video object segmentation by tracking regions," in *ICCV*, 2009.

[12] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *CVPR*, 2009.

[13] X. Ren and J. Malik, "Tracking as repeated figure/ground segmentation," in *CVPR*, 2007.

[14] S. Liu, G. Dong, C. Yan, and S. Ong, "Video segmentation: Propagation, validation and aggregation of a preceding graph," in *CVPR*, 2008.

[15] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *ECCV*, 2010.

[16] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *CVPR*, 2010.

[17] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video snapcut: robust video object cutout using localized classifiers," in *SIGGRAPH*, 2009.

[18] D. Tarlow, R. Adams, and R. Zemel, "Randomized optimum models for structured prediction," in *AISTATS*, 2012.

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

[20] X. Boix, J. Gonfaus, J. Van De Weijer, A. Bagdanov, J. Serrat, and J. Gonzalez, "Harmony potentials," *IJCV*, 2012.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels," *TPAMI*, 2012.

[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

[23] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *ICML*, 2011.

[24] X. Boix, G. Roig, and L. Van Gool, "Nested sparse quantization for efficient feature coding," in *ECCV*, 2012.

[25] Y. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in vision algorithms," in *NIPS*, 2010.

[26] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *TPAMI*, 2012.

[27] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzinski, C. Strecha, and P. Fua, "BRIEF: Computing a Local Binary Descriptor Very Fast," *TPAMI*, 2011.