Online Learning for Scene Segmentation With Laser-Constrained CRFs

Charika De Alvis Lionel Ott Fabio Ramos

***Abstract*— Scene understanding is a crucial requirement for robot navigation. Conditional Random Fields (CRF) are commonly used to solve the scene labelling problem since they represent contextual information efficiently and provide efficient inference methods. However, when a robot navigates through an unknown environment, it is often necessary to adjust the parameters of the CRF online to maintain the same level of accuracy under changes no predicted during the training phase. Online parameter learning can be challenging since ground truth information is not available for newly encountered scenes. To address this issue, this paper proposes a stochastic gradient descent (SGD) method to learn the parameters of a constrained CRF (cCRF) in an online fashion. By leveraging the information from laser scans and image data the complexity of the labelling problem can be significantly reduced. The parameters are estimated by optimising a novel loss function that takes into account highly confident labels as a reference while eliminating the need for manual labelling. These labels are obtained purely based on the information from camera and laser sensors, in a self-supervised manner. Sensor data is pre-processed using methods such as convolutional nets, discriminant analysis, and Euclidean distance based clustering to extract reference labels. We show that this online parameter learning is robust to changes in the data distribution by selecting the learning rate appropriately. Experimental results are presented on the KITTI data set demonstrating the benefits of online CRF training.**

I. Introduction

Scene understanding is an important skill for many robotic tasks. It provides the foundation, which allows a robot to perceive and reason about its environment. For navigation in urban settings, such information is crucial for safety, as it allows the robot to identify the areas that can be a risk due to the presence of dynamic objects. Commonly, CRF models are used to perform scene labelling, since they excel at integrating local classifiers and spatial smoothness. However, efficient combination of this information is challenging.

Particularly in autonomous navigation, where the robot's environment is continuously changing the efficient combination of features is very important. Adaptively and continually learning the CRF parameters is therefore coupled with the current distribution of data. However, CRF parameter learning can be painstaking due to complex correlations between variables and the cost involved with inference. Due to the enormous cost associated with computing the CRF objective function, stochastic gradient methods that use a gradient calculated at a single point or small subset of the data, instead of the actual gradient, is an appealing alternative. As a result, Stochastic Gradient Descent (SGD) algorithms are widely used in online learning.

Charika De Alvis, Lionel Ott and Fabio Ramos are with the School of Information Technologies, The University of Sydney, Australia.

Training of CRF is commonly conducted with fully labelled images. In some cases partially labelled images are used to train CRF since it also helps to overcome issues such as parameter over fitting and over-estimation. However in autonomous navigation the learning occurs as the new data encounters where no ground truth data is available. In this scenario parameter learning is not a trivial problem.

The purpose of this paper is to propose an online parameter-learning framework for CRF based scene-labelling models by eliminating the use of manually labelled images and robustly continue parameter learning in changing environments. In this setup, we are searching for the best set of parameters for CRF model to accurately predict the labels for the current image. In other words, we are interested in finding the best local estimate rather than a global optimum for all the encountered images in the past. We use information from the camera and laser sensors as a reference to compute the loss of wrong labelling. By minimising this loss, we obtain the parameters for the current context. We use an SGD-based approach for the optimisation as it facilitates getting the best parameter set based on the recent data. We demonstrate the performance of the online parameter learning on constrained CRF (cCRF) model proposed in [2]. We use the real world street scene data in KITTI [8]. To summarise the main contributions of the paper are:

1) Developing a model to learn CRF parameters by eliminating the need for ground truth labels. The model derives reference labels by pre-processing the sensor information.
2) Using stochastic gradient based method to update the parameters while making the method robust to non-stationary data in the long term deployments.
3) Learning parameters of cCRF model in a online fashion using the techniques (1) and (2) to make it robust when labelling continuous streams of data

II. Related Work

A number of approaches have proposed to efficiently estimate the parameters for CRF models. Verbeek *et al.* [20] introduce a method for learning CRFs by marginalising out the variables with unknown labels and by maximising the log-likelihood of the variables with known labels using gradient ascent. Tsuboi *et al.* [18] also present a similar framework for training CRFs using a partially annotated corpora to conduct natural language processing. In [9], authors develop a hybrid model for exploiting incompletely labelled data that combines a generative topic model for image appearance with discriminative label prediction. In [10], authors propose a method for parameter learning in dense random fields. This

method uses information about the dependencies between parameters by learning them jointly. The loss are functions computed bases on mean field marginal. These methods target the offline learning of the CRF parameters.

For large scale learning problems it need algorithms that can scale favourably. Due to high cost associated with CRF inference SGD methods are commonly used instead of batch learning methods in online settings. The momentum method [14] is commonly used to help SGD to accelerate in relevant directions and dampen oscillations. Selecting an ideal learning rate for SGD can be challenging. ADAGRAD [4] is one popular method of updating the learning rates since it only uses first order assumptions. However, the method tends to have properties of second order methods and annealing implicitly. ADADELTA [22] is a recent improvement which assists SGD to overcome the sensitivity to the hyper parameter selection. Further, it also prevents continual decaying of the learning rates and facilitates to escape from local minima by allowing the learning rate to progress. However in most of the problems the SGD has a slower convergence rate. To overcome this issue Schmidt *et al.*[16] apply the stochastic average gradient (SAG) algorithm which combine the characteristics of deterministic and stochastic models to train CRFs. They show that this algorithm converges with a less number of iteration than SGD. However given this advantage still it may be difficult to apply SAG algorithm for models with complex features that associate with a greater number of labels and also it confine to problems with finite number of training examples.

In our research, we are interested in online learning for autonomous navigation. Schraudolph *et al.* [17] propose a scalable, stochastic quasi-Newton method for online convex optimisation. In online settings when new data appear without a prior knowledge loss functions are not guaranteed to be convex all the time. Hence non-convex optimisation is a matter of interest. Schaul *et al* [15] propose a method to automatically tune learning rates to minimise the expected error in the current situation. The framework performs well in non-convex problems. The method is based on local gradient variations across samples. In this framework, learning rates have the freedom to progress or diminish to make it robust for non-stationary problems. The framework is aimed for solving computer vision problems. Our framework also uses a similar technique to change learning rates to adapt changing data distributions but focus on exploiting the robot sensor data.

Fathi *et al.* [6] propose an incremental self-training algorithm where they iteratively label the least uncertain frame and update similarity metrics. This self-training video segmentation provides higher accuracy for foreground identification problems. The approach of [13] consists of combined self-learning algorithm for ground detection. The system automatically learns to ally salient features that are extracted from sensor data in correspondence to the ground class. New observations are labelled by outlier rejection using the past data. Vijayanarasimhan *et al.* [21] demonstrate a method to reduce human effort in video annotation. They choose k number of frames for manual labelling to ensure that automatic pixel level label propagation would occur with minimal expected error. Here they minimise the effort required for labelling and correcting propagation errors. All these methods require some amount of labelling of the data, which is difficult to obtain in real time navigation tasks. Our framework omits the need to use labelled data in the learning process, which instead try to exploit the existing sensor information and educate the parameters to adjust meaningfully.

III. Background

Figure 1 shows a overview of the proposed framework. We use this framework to learn the parameters of the cCRF model [2] in a online fashion. cCRF conducts scene segmentation by enforcing a set of global constraints on the optimisation which makes it more computationally efficient. Further it also has a high accuracy in scene segmentation and requires only unary and pairwise potential terms. The following section summarises the formulation of the cCRF model.

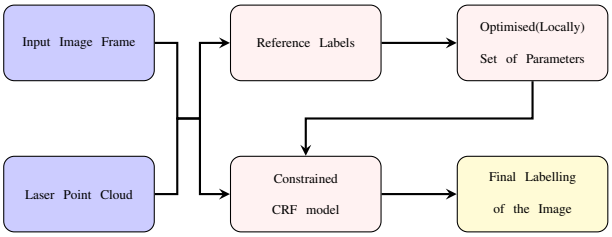

Fig. 1: Overview of the online parameter learning for cCRF model

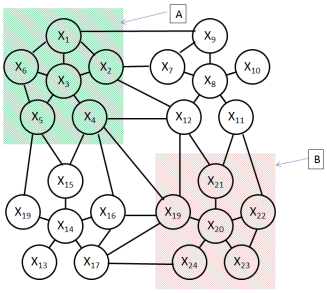

Fig. 2: Example of the type of CRF graph used in this paper. Pairwise potentials are indicated by the edges, while the additional constraints are indicated by the two shaded areas, A and B. These areas encode sets of nodes which are required to be assigned the same label.

A. cCRF Model

A graphical representation of the cCRF model presented in [2] is depicted in Figure 2, where circles and edges denote nodes and connections between nodes respectively. Each node corresponds to a super pixel [1] in the image and the neighbouring nodes are connected. The two sets of nodes coloured identically in Figure 2 represent sets of nodes constrained to take the same label. Where S is the set of super pixels in the image. $X = \{x_1, x_2, \ldots, x_N\}$ is the set of discrete random variables, where x_i corresponds to the label

prediction of super pixel/node i. Each super pixel can take one of the output labels $L = \{1, \ldots, n\}$, where n denotes the number of classes.

The solution to this CRF is obtained as a maximum a posteriori (MAP) estimation of the conditional log-likelihood;

$$\log P(X \mid S) = \sum_{i \in S} \phi_i(x_i) + \sum_{i \in S, j \in \mathcal{N}(i)} \psi_{ij}(x_i, x_j) - Z(S). \quad (1)$$

The potential functions of the CRF are indicated as $\phi_i(x_i)$ for the unary potential and $\psi_{ij}(x_i, x_j)$ for the pairwise potential correspond to each super pixel i and each of its neighbours $j \in \mathcal{N}(i)$. $Z(S)$ denotes the normalising term. To solve the inference problem $P(X|S)$ efficiently Zhang *et al* [23] propose a quadratic programming (QP) relaxation. To improve this label assignment Charika *et al* [2] add global constraints to the QP problem. This constrained quadratic programming model is denoted as follows,

$$\text{maximise} \quad \sum_{i \in S} \sum_{p \in L} \theta_p^{unary} \phi_i(x_i^p) \mu_i(x_i^p) + \sum_{\substack{i \in S \\ j \in N(i)}} \sum_{p,q \in L} \theta_{pq}^{pair} \psi_{ij}(x_i^p, x_j^q) \mu_i(x_i^p) \mu_j(x_j^q) \quad (2a)$$

$$\text{subject to} \quad \sum_{p \in L} \mu_i(x_i^p) = 1 \quad \forall i \quad (2b)$$

$$\sum_{i,j \in C_k} \sum_{p \in L} \mu_i(x_i^p) - \mu_j(x_j^p) = 0 \quad \forall C_k \in C \quad (2c)$$

$$0 \leq \mu_i(x_i^p) \leq 1 \quad \forall i, p, \quad (2d)$$

where,

$$\theta_{pq}^{pair} = \begin{cases} \theta_p^{\text{ondiag}} & \text{if } p = q \; p, q \in L \\ \theta_p^{\text{offdiag}} & \text{otherwise} \end{cases}, \quad (3a)$$

$$\Theta = [\theta_1^{\text{unary}}, ... \theta_n^{\text{unary}}, \theta_1^{\text{ondiag}}, ... \theta_n^{\text{ondiag}}, \theta_1^{\text{offdiag}}, ... \theta_n^{\text{offdiag}}]. \quad (3b)$$

This cCRF model has parameters corresponding to unary and pairwise terms. The relaxed indicator variable $\mu(x_i^p)$ denotes the probability of random variable x_i taking label p. Eq. (2c) enforces that all pairs of points i and j in a constraint set $C_k \in C$ are assigned the same label. These global constraints obtained from additional sensor data. Charika *et al* [2] reformulate the problem by enforcing the global constraints implicitly in the optimisation problem which results in a large dimensional reduction in the QP problem. Inference of this reduced problem is done by the gradient based approach proposed by Zhang *et al*.

IV. Online Learning

In this section we present our framework which permits us to optimise the parameters Θ of the unary and pairwise potentials of the cCRF model in an online fashion. We optimise the loss function, detailed next, using stochastic gradient descent (SGD) which allows for fast and continues updates of the parameters.

A. Loss Function

Our goal is to minimise the difference between the reference labels Γ extracted in a self-supervised manner using sensor information and cCRF label prediction $\mathbf{r}$ of the corresponding super pixels in S by selecting the optimal CRF parameters Θ,

$$\Theta^* = \arg\min_{\Theta} l(\Gamma, \mathbf{r}). \quad (4)$$

where Θ^* is the set of optimal parameters we wish to find and l is the loss function we need to optimise. cCRF label prediction $\mathbf{r}_{ip} = \mu_i(x_i^p)$ where $i \in S$ and $p \in L$.

Ideally we would compare the predicted result to ground truth labels, as is typically done in parameter learning. However, as we operate in an online setting we do not have access to such ground truth labels for the data we observe. Therefore we extract labels for super pixels where we are highly confident about the label purely based on laser point clusters, fully convolutional net (FCN) [11] classifier results, and pseudo linear discriminant analysis classifier (pLDA) [12] results. As such in each frame we process we will have a varying number of super pixels with reliable reference labels at our disposal. Putting all this together we obtain the following loss function:

$$l = l_{\text{agree}} + l_{\text{differ}} + l_{\text{laser}}. \quad (5)$$

Where,

$$l_{\text{agree}} = \sum_{S_j \in S_{\text{agree}}} \lambda_j \sum_{i \in S_j} ||\mathbf{r}_i - \Gamma_i||^2, \quad S_{\text{agree}} = [S_1, .., S_n]. \quad (6)$$

l_{agree} provides a measure of deviation from reference labels, where λ_j is the weight for the loss component of each class. S_j denotes the set of superpixels which we are confident the true label is j based on the classifiers and laser point clusters, i.e. super pixels, covered by a point cloud segment that does not belong to the ground plane and both classifiers predict the label to be a vehicle will be added to the set of reference labels to use in the computation of the loss function. l_{differ} is used in cases where we have knowledge that a certain label assignment is not possible, i.e. a super pixel that is observed by the laser cannot be sky. Here we add a loss if the cCRF predictions assign a label to a less probable class (based on the knowledge from laser clusters and classifiers),

$$l_{\text{differ}} = \sum_{\mathcal{S}_j \in \mathcal{S}_{\text{differ}}} \alpha_j \sum_{i \in \mathcal{S}_j} ||\mathbf{r}_i \cdot \Gamma_i||^2, \quad \mathcal{S}_{\text{differ}} = \{\mathcal{S}_1, .., \mathcal{S}_n\}, \quad (7)$$

where α_j is the weight for the loss component of each class. $\mathcal{S}_j$ denotes the set of super pixels which we are confident the true label is not j based on the classifiers and laser point clusters. Finally for parts where we have point cloud segments we assign a loss, l_{laser}, if cCRF prediction violates the label consistency obtained by the laser segments,

$$l_{\text{laser}} = \lambda_l \sum_{\mathcal{C}_j \in \mathcal{C}} \sum_{i \in \mathcal{C}_j} ||\mathbf{r}_i - \mathbf{r}_{i+1}||^2. \quad (8)$$

Putting these parts together with a regularizer to prevent overfitting we obtain the following optimisation problem:

$$\Theta^* = \arg\min_{\Theta} \sum_k l + ||\exp(\Theta)||^2, \quad (9)$$

where each k is a new image. This type of function is amenable to optimisation using stochastic gradient descent. For our method we propose to use ADAGRAD which is described in the next section.

B. Stochastic Learning

As we operate in an online setting where we continuously obtain new observations standard batch gradient optimisation methods are not applicable due to he unbounded size of the data to be processed. As such we use stochastic gradient descent (SGD), which operates on a single observation at a time, to optimise the parameters using the loss function presented in Eq. (9).

For each image (iteration) we compute a stochastic gradient with which to update the parameter vector Θ. To this end we form a mini-batch composed of the last M images and perform M parameter update steps. The value of M is dependent on the rate at which images are received, higher rates allows us to use larger values of M. The values of Θ obtained in this way are adapted to the current context of the scene, however, also retain information from the past. As the learning rate has a big impact on the speed of convergence and quality of the obtained result we employ ADAGRAD [4] which uses individual learning rates for each parameter that change based on past data. The basic equations of ADAGRAD have the following form:

$$G = \sum_t g_t g_t^T \quad (10)$$

where $g_t = \bigtriangledown l(\Gamma, \mathbf{r})$ is the gradient at iteration t. With this we can update the parameter set Θ as follows:

$$\Theta := \Theta - \eta Diag(G)^{-1/2} \circ g, \quad (11)$$

where η is the global learning rate, g the current gradient. While ADAGRAD works well in typical large scale problems there are some drawbacks when using it in an online setting. The main one is that the entire gradient value history is accumulated which results in a continuously decreasing step size. In an online settings this means that at some point the parameters would no longer adapt to changes in the environment. One possible solution is to use a constant fixed learning rate which would always allow for changes in the environment to be reflected in the parameters. However, selecting a suitable fixed learning rate is not trivial and would require a lot of testing for different scenarios which clearly isn't ideal. So in order to have the good learning rates of ADAGRAD while still being able to adapt to changes we adopt a procedure similar to that of [15].

The basic idea is to have a decaying learning rate, but at opportune moments increase this learning rate again to allow quicker adaptation. In our case once the learning rate has become sufficiently small for a number of iterations we set $G = 0$ which discards all previously accumulated gradient information. This effectively increases the learning rate and allows the optimiser to adapt to changes if necessary. In the case that the distribution has changed the gradient will be non-zero and pull the solution to a different local minimum. Similarly, if the distribution hasn't changed the gradients will be close to zero and the algorithm will not change the parameters.

Algorithm 1: Online Learning Algorithm

```
// cCRF(..)- MAP estimation of cCRF model
// t - Iteration number
// w - Image frame index
// δ, υ - Thresh hold values
// G - Accumulated gradient
// Θ_w - Parameter set correspond to w^th image frame
Input: η-Global learning rate , I - Input image , M - Mini batch size
Output: Label assignment X
// Initialisation
1  w = M + 1 , G = 0 , Θ_w = [1]_{1×21}   ∀w ∈ [1, ..M]
// SGD parameter optimisation
2  while Images available do
       // Select past M frames and shuffle
3      foreach ∀t ∈ {w − M, .., w} do
4          r ← cCRF(Θ, I_t)
5          Γ ← self supervised reference labels of image I_t
6          loss=l(Γ, r)
           // gradient of the loss function
7          g ← dl/dΘ
           // gradient accumulation
8          G ← G + gg^T
           // updating the parameters
9          Θ_w ← Θ_w − ηDiag(G)^{−1/2} · g
10     end
       // Decide when to reset the step size
11     if ∀τ ∈ [t : t − υ]; abs(Θ_τ − Θ_{τ−1}) < δ then
12         G = 0;
13     end
14     Θ_{w+1} ← Θ_w
15     X ← cCRF(Θ_{w+1}, I_{w+1})
16     w ← w + 1
17 end
18 return X
```

An overview of the steps involved in our algorithm are summarised in Algorithm 1. With each new image the last M images, typically 10 to 20, are used to compute the gradients of the parameters Θ using the loss function (lines 4 to 9). Next we decide whether or not to reset the step size which allows us keep adapting to changes (lines 11 to 13). Finally, we update the parameter set Θ and obtain the segmentation results before processing the next image (lines 14 to 16). Our experiments we have considered 7 object classes (n=7) which resulted in 21 parameters.

V. Experiments

In this section we present experimental evaluation of our proposed framework for online learning of CRF parameters. The results compare the results obtained using cCRF with fixed parameters with those obtained using cCRF

α_1	α_2	α_3	α_4	α_5	α_6	
0.47	8.00	0.00	0.00	0.20	0.30	
λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_l
1.50	0.47	0.50	0.75	0.50	3.00	2.00

TABLE I: Overview of the loss function parameters.

with adaptive parameters. We use the KITTI dataset [8] as it provides typical image and laser scanner data collected in urban environments. The data which was collected in the city of Karlsruhe using a vehicle equipped withe cameras and a Velodyne laser scanner provides a variety of scenes and environmental conditions. Test set includes images from `drive_0021`, `drive_0043`, `drive_0071`, `drive_0038`, `drive_0093` and `drive_0095`.

A. Constrained Conditional Random Field

Our goal is to segment the images into the following seven classes: pedestrians and cyclists, ground, vegetation, buildings, sky, vehicles, and unknown. The unary potentials ϕ_i of the CRF are obtained from the combination of the posteriors of two classifiers. A pseudo linear discriminant analysis classifier [12] trained on the KITTI dataset using HSV colour histograms, RGB Hog features [3], and pixel coordinates corresponding to each super pixel. And a FCN classifier based on the pre-trained `pascal-fcn32s-dag` net [19] trained using the pascal [5] dataset. Figure 4 shows the classifier outputs for a raw image. The weighted average of the posteriors of these two classifiers is used as the unary potentials. For the pairwise potentials the following simple function is used:

$$\psi_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0.01 & \text{otherwise} \end{cases} \tag{12}$$

The constraints required by both the constrained CRF as well as the self-supervised labelling process are extracted from the Velodyne scans using a simple process. Firs the ground plane is removed using RANSAC [7] to find the largest ground plane. The remaining points are clustered using an Euclidean distance based algorithm. Of the resulting clusters only those with at least a certain number of points are retained. These 3D segments are then mapped, using the extrinsic calibration data provided by the KITTI dataset, into the image space to obtain the corresponding super pixels.

The seven classes associates with, 14 parameters to control the pairwise potentials and 7 to control the unary potentials. All nodes in the CRF use the same set of parameters. The gradient of the loss function Eq. (4) is calculated using central finite differences [16].

The weights of the loss function λ and α were chosen through a grid search followed by a fine tuning on `drive_0091`. These values are summarized in Table I. The base learning rate was selected as $\eta = 0.037$ in a similar manner.

B. Results

In the following we present results comparing cCRF using fixed parameters and cCRF using parameters that are adapted online using our proposed method. An overview of the typical behaviour and performance of the proposed algorithm is shown in Figure 3. The top image demonstrates how the online adaptive cCRF maintains a higher overall accuracy in comparison to cCRF using fixed parameters. This is clearly visible in the areas where cCRF has drops in accuracy which the online cCRF manages to avoid as it adapts to the changes and as a result doesn't drop as much in terms of accuracy. The middle and bottom image evaluate the accuracy on the parts of the image for which we have obtained labels (middle) and those where we have had no label information (bottom). As to be expected the result for areas where we have labels is better then for those where we lack label information, however, the difference is relatively small. Overall the shapes and trends are quite similar which is a good indication that the parameter training done on the labelled parts influences the parameters of classes without labels in a positive way. One interesting case are the two drops in performance around the frame #200 and #350. In the first instance this drop is present in both labelled and unlabelled data and as a result the online cCRF manages to mitigate it. By contrast the second instance only occurs in the unlabelled part of the data and as such no parameter adaptation happens because of it and both the online cCRF and fixed cCRF reduce in accuracy. This again demonstrates that parameters updated based on the labelled parts of the data improves the performance in areas where we have not obtained labels. Figure 5 contains the image frames correspond to the marked points A,B,C,D in the top graph of Figure 3.The results of online CQP avoid the errors in CQP solution occurred due to illumination and noise.

The same type of improvements can be observed in other datasets. Figure 6 shows the relative change in accuracy between cCRF and online cCRF, i.e. a positive value indicates that online cCRF is performing better then cCRF using fixed parameters. From these plots we can see the constant gain in accuracy where the spikes stem from sudden drops in accuracy in cCRF which online cCRF manages to mitigate. These results are also verified in the comparison of several performance metrics on multiple datasets in Table II. The table shows how online cCRF consistently improves on the results obtained by cCRF. This improvement is typically in the 2% to 3% range, but in a few cases the gain is as much as 6%.

Next we are going to look at the per class performance to see the impact online cCRF has on those. Looking at Figure 7 we can see that for very simple classes such as "ground" there is barely any improvement. For more complex and varied classes this changes. In the case of the "pedestrian and cyclists" class there is mostly no change, however, when cCRF makes large errors the online cCRF method maintains good accuracy. Looking at the "vegetation" and "buildings" classes we can see that online cCRF has a somewhat

Quality Measure	Average Precision		Average Recall		Average Accuracy		F1 Score	
Method	cCRF	Online cCRF	cCRF	Online cCRF	cCRF	Online cCRF	cCRF	Online cCRF
Dataset0071	0.8440	0.8676	0.8793	0.8987	0.9493	0.9562	0.8237	0.8481
Dataset0095	0.8501	0.8938	0.9350	0.9393	0.9496	0.9642	0.8435	0.8884
Dataset0038	0.7454	0.7860	0.7135	0.7780	0.9317	0.9441	0.7284	0.7814
Dataset0093	0.8534	0.8767	0.8390	0.8624	0.9503	0.9601	0.8458	0.8692

TABLE II: Quantitative comparison of cCRF and online cCRF on different dataset. Online cCRF consistently improves on the results of cCRF in varied datasets, such as Dataset0071 which has a high concentration of pedestrians to Dataset0095 which contains a large amount of vehicles.

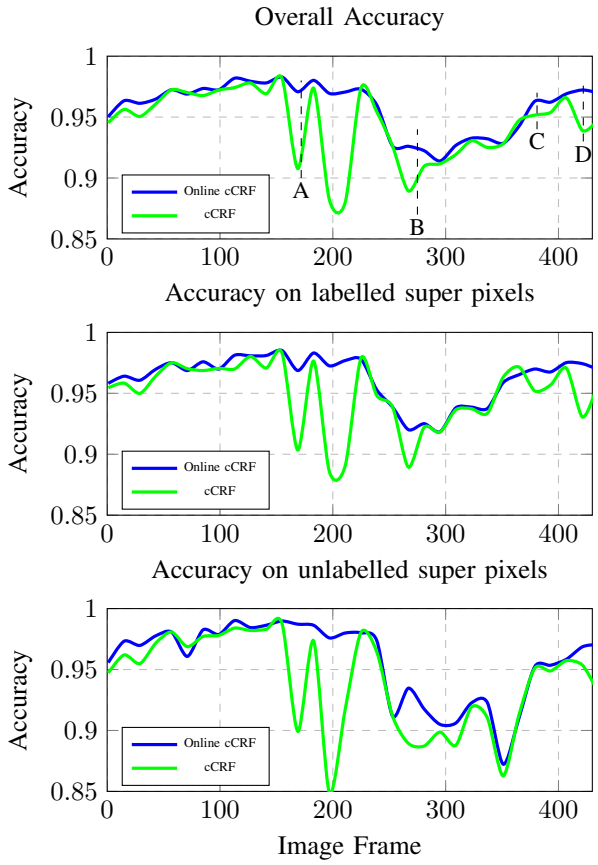

Fig. 3: Accuracy on a per frame basis for the `drive_0093` dataset from KITTI. (Top) overall accuracy of each frame, (middle) accuracy of the super pixels for which we extracted labels in a self-supervised manner, and (bottom) accuracy for super pixels without label information. Overall the online cCRF is able to adapt the parameters to prevent drastic reduction in accuracy. Comparing the (middle) and (bottom) graphs one can see that even though the parameters are learned only on data from the (middle) the changes have a positive impact on the (bottom) graph.

smoother curve while exhibiting a positive accuracy offset over cCRF. These impressions are also verified by the numerical evaluation presented in Table III for `drive_0093`. For hard classes such as "Cyclists & Pedestrians" the precision does not improving, however, recall improves significantly which also reflects in the F1 score. Depending on the class some metrics remain unchanged while others gain and as a result the F1 score improves across the board. As such the online cCRF method manages to improve on the challenging metrics for each class without degrading others.

While during typical autonomous navigation the environment changes smoothly rather then abruptly we evaluated the ability of our proposed method to quickly adapt to changes in the data. To this end we selected two very different datasets, `drive_0093` which contains mainly vehicles and `drive_0071` which has data captured in a pedestrian zone. These two datasets were processed one after the other as if they were one continuous data stream. In Figure 8 we show the evolution of the unary potential parameters of online cCRF (top) and on-diagonal pairwise parameters (bottom)

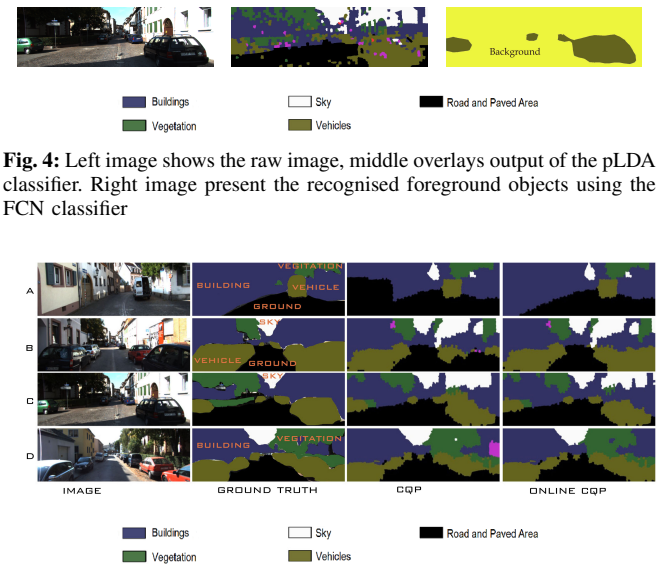

Fig. 4: Left image shows the raw image, middle overlays output of the pLDA classifier. Right image present the recognised foreground objects using the FCN classifier

Fig. 5: Example images for the quality improvement in online CRF

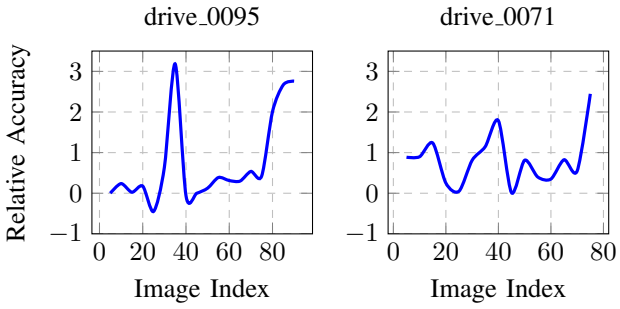

Fig. 6: The plots show the relative accuracy, i.e. difference in absolute accuracy values, between cCRF using fixed parameters and online cCRF. A positive value indicates that the accuracy of online cCRF is better then that of cCRF. We can see how online cCRF is outperforming cCRF in almost all cases. Big spikes in the relative accuracy can be explained by a drop in accuracy of cCRF that online cCRF managed to adapt to in time.

Quality Measure	Average Precision		Average Recall		Average Accuracy		F1 Score	
Method	cCRF	Online cCRF	cCRF	Online cCRF	cCRF	Online cCRF	cCRF	Online cCRF
Cyclists & Pedestrians	0.8066	0.7994	0.5797	0.7279	0.9233	0.9346	0.5184	0.6375
Ground	0.7223	0.7805	0.9095	0.9379	0.9145	0.9363	0.8335	0.8687
Vegetation	0.8923	0.8949	0.8830	0.8658	0.9705	0.9707	0.8622	0.8624
Buildings	0.9527	0.9515	0.9210	0.9219	0.9050	0.9124	0.8768	0.8888
Sky	0.6674	0.6911	0.8885	0.8960	0.9868	0.9877	0.7435	0.7521
Vehicle	0.8701	0.8987	0.8801	0.8955	0.9123	0.9345	0.8751	0.8955

TABLE III: Class wise accuracy, precision, recall, and F1 score for cCRF and online cCRF on the `drive_0093` dataset. Different metrics are improved for different classes which is dependent on what makes a class hard to classify correctly. However, across the board the F1 score increases, indicating that online cCRF manages to improve on hard aspects of the classification without sacrificing other areas.

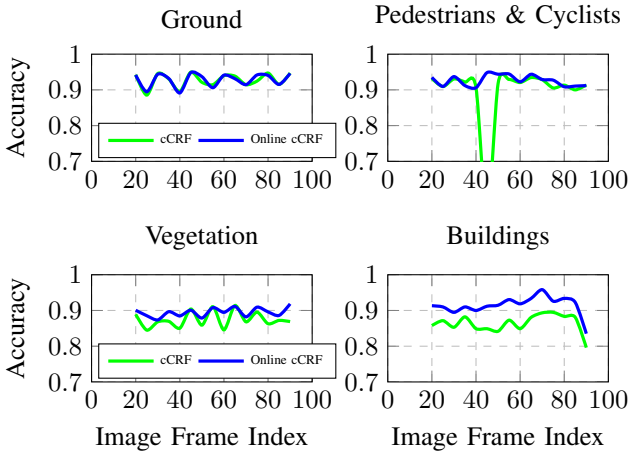

Fig. 7: Accuracy of cCRF and online cCRF on a per class basis for the `drive_0038` dataset. For easy classes there is little difference, however, in more complex ones we can see online cCRF retaining good accuracy when cCRF drops significantly as seen in "Pedestrians & Cyclists" or has a constant performance offset as in the "Buildings" class.

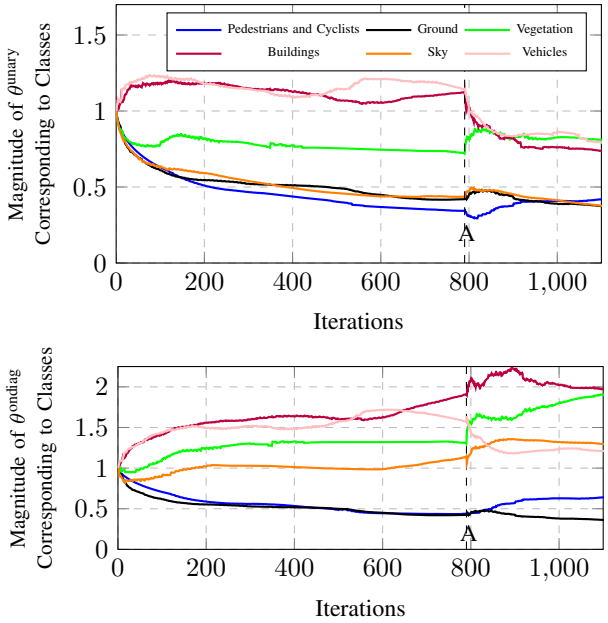

Fig. 8: The top and bottom plots show the change of parameters correspond to unary and pairwise potentials with adaptive learning. The test is done for `drive_0093`. At iteration A data sequence `drive_0071` is fed to the framework which has different lightning conditions and class distribution than the previous one. Plots clearly depict that after this sudden change on input data, parameters dramatically change to adapt the situation

as we process the data. All parameters start with a value of 1 and we can see how they quickly move to mostly stable values different from 1. Then around iteration 800 the first dataset ends and the second one starts being processed. We can see abrupt jumps in the values indicating that the SGD method is able to quickly change parameters if needed. After this short period of rapid changes all parameters settle again. The actual direction in which the parameter values move is not necessarily indicative of the scene composition as the parameter interact in complex ways inside the cCRF method itself.

The importance of being able to quickly adapt to changes, even if this is a rare occurrence, is demonstrated in Figure 9 which compares the accuracy of the first 50 frames after we switch the datasets. We compare the results of cCRF using the same fixed parameters, online cCRF which contentiously adapts its parameters and partial online cCRF which adapts the parameters until the dataset changes, i.e. the parameters at point "A" in Figure 8 are used. This allows us to evaluate how important the ability to adapt quickly is. We can see that both online cCRF methods outperform cCRF which is in line with the previous results. The interesting part is the comparison of the two online cCRF methods. In several areas we can observe that the lack of adaptability results in degraded performance, for example around frame 30 and 40. As such being able to react quickly to changes in the environment is important to prevent errors from accumulating over time.

All computations were performed on an Intel Core-i5 3.20GHz processor with MATLAB implementations of the algorithms. Each parameter update requires 70 ms. As the parameter updates are independent of the segmentation itself it is possible to perform the segmentation at a higher frequency then the parameter updates. Furthermore, the number of images M considered in a single update step can be chosen in a wide range. As we can see in Figure 10 the performance stays very stable with 10 or more images used. This means that longer range information, from older images, does not negatively impact the adaptation capability of the algorithm.

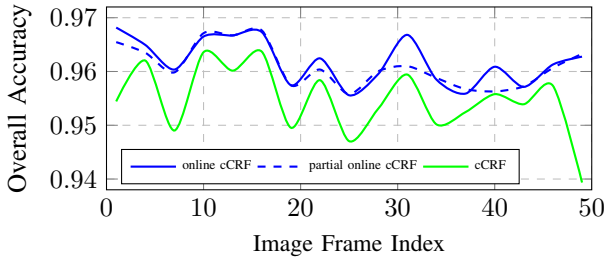

Fig. 9: Accuracy of the first 50 frames after the new dataset was introduced. Online cCRF continues to adapt, while partial online cCRF continues to use the parameters used at the end of the first dataset while cCRF uses the same initial parameters. We can see how the continued adaptation allows online cCRF to improve over the partial online cCRF. As seen previously both versions of online cCRF outperform cCRF using fixed initial parameters.

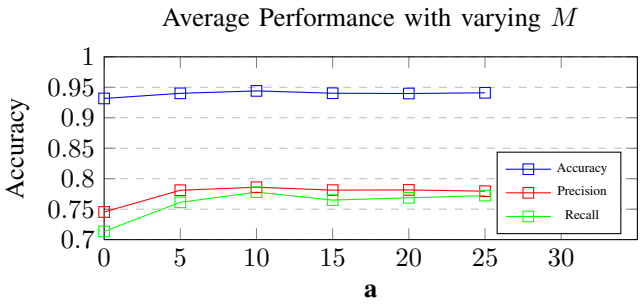

Fig. 10: The plots shows the quality of the segmentation of the image sequence `drive_0038` with online cCRF with varying number of images M considered in each update step.

VI. Conclusion

In this paper we presented a method that learns the parameters of the unary and pairwise potentials of a CRF in an online manner. This enables the algorithm to adapt the parameters based on the current situation which is advantageous in a life-long learning scenario where it the environment is expected to change over time. This is achieved by formulating the selection of the optimal parameters as a loss function using reference labels that are obtained in a self-supervised manner. This loss function is updated efficiently using stochastic gradient descent with continuously adapting learning rates. In experiments conducted using data from the KITTI dataset we demonstrate the benefit in regards of scene segmentation performance of a CRF that continuously adapts it's parameters over one with fixed parameters. Furthermore, we demonstrated that the proposed method can quickly adapt to changes in the environment.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sabine. SLIC Superpixels. Technical report, EPFL, 2010.

[2] C. Alvis, L . Ott, and F. Ramos. Urban scene segmentation with laser-constrained crfs. In *International Conference On Intelligent Robots and Systems*, 2016.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[4] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[5] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[6] A. Fathi, M. Balcan, X. Ren, and J. Rehg. Combining self training and active learning for video segmentation. In *Eds. Jesse Hoey, Stephen McKenna and Emanuele Trucco, In Proceedings of the British Machine Vision Conference (BMVC 2011)*, volume 29, pages 78–1, 2011.

[7] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics : The KITTI Dataset. *The International Journal of Robotics Research*, 2011.

[9] X. He and R. Zemel. Learning hybrid models for image annotation with partially labeled data. In *Advances in Neural Information Processing Systems*, 2009.

[10] P Krähenbühl and V Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pages 513–521, 2013.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[12] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher Discriminant Analysis with Kernels. In *Proc. of the IEEE Signal Processing Society Workshop Neural Networks for Signal Processing*, 1999.

[13] A. Milella and G. Reina. Adaptive multi-sensor perception for driving automation in outdoor contexts. *International Journal of Advanced Robotic Systems*, 11, 2014.

[14] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[15] T. Schaul, S. Zhang, and Y. LeCun. No more pesky learning rates. *ICML (3)*, 28:343–351, 2013.

[16] M. Schmidt, R. Babanezhad, M. Ahmed, A Defazio, A. Clifton, and A. Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. *arXiv preprint arXiv:1504.04406*, 2015.

[17] N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 436–443, 2007.

[18] Y. Tsuboi, H. Kashima, H Oda, S. Mori, and Y. Matsumoto. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 897–904. Association for Computational Linguistics, 2008.

[19] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the ACM International Conference on Multimedia*, 2015.

[20] J. Verbeek and W. Triggs. Scene segmentation with crfs learned from partially labeled images. In *NIPS 2007-Advances in Neural Information Processing Systems*, volume 20, pages 1553–1560. MIT Press, 2008.

[21] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *Computer Vision–ECCV 2012*, pages 496–509. Springer, 2012.

[22] M. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[23] Y. Zhang and T. Chen. Efficient inference for fully-connected CRFs with stationarity. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.