Sequential Gaussian Mixture Models for Two-Level Conditional Random Fields

Sergey Kosov, Franz Rottensteiner, and Christian Heipke

Institute of Photogrammetry and GeoInformation (IPI), Leibniz Universität Hanover

Abstract. Conditional Random Fields are among the most popular techniques for image labelling because of their flexibility in modelling dependencies between the labels and the image features. This paper addresses the problem of efficient classification of partially occluded objects. For this purpose we propose a novel Gaussian Mixture Model based on a sequential training procedure, in combination with multi-level CRF-framework. Our approach is evaluated on urban aerial images. It is shown to increase the classification accuracy in occluded areas by up to 14,4%.

1 Introduction

Labeling of image pixels is a classical problem in pattern recognition. Probabilistic models of context with the goal of achieving a smooth classification result have been increasingly used to model dependencies between neighbouring image sites. A recent comparison of smooth labelling techniques for remote sensing imagery has shown that this is essential, with Conditional Random Fields (CRF) [12] performing best among the compared techniqes [18].

CRF have been applied successfully to many labelling tasks in computer vision and remote sensing [12, 18, 19, 22], but they have problems with proper labelling of partially occluded objects. Occlusion of roads by trees or cars has been known to be a major problem of road extraction from remote sensing imagery for a long time. Model-based techniques have tried to overcome this problem by treating such objects as context objects in an ad-hoc manner [8], but a sound statistical treatment of the problem is still missing.

Previous work on the recognition of partially occluded objects includes [13], where the objects in the scene are represented as an assembly of parts. The method is robust to the cases where some parts are occluded and, thus, can predict labels for occluded parts from neighbouring unoccluded sites. However, it can only handle small occlusions, and it does not consider the relations between the occluded and the occlusion objects. We handle this problem by using a two-layered CRF (*tCRF*) [10], which explicitly models *two* class labels for each image site, one for the occluded object and one for the occluding one. In this way, the 3D structure of the scene is explicitly considered in the structure of the CRF. Labelling is supported by depth information obtained from image matching.

There have been a few attempts to include multiple layers of class labels in CRFs [11, 19, 22, 23]. However, these methods cannot be applied to our problem. Firstly, they use part-based models where the additional layer does not explicitly refer to occlusions, but encodes another label structure. Furthermore, many of

them rely on object detectors. Thirdly, in vertical views (typical in remote sensing), models based on the absolute position or orientation in the image cannot be applied because there is no natural definition of a direction of reference such as the vertical in images of street scenes; applying such models would imply learning models of the distribution of features relative to the nadir point of an image or to the North direction. None of these publications use depth information as an additional cue to deal with occlusions.

tCRF does not need additional foreground object-detectors in order to separate the foreground from the background level. The information from neighbouring unoccluded objects as well as from the occluding layer will contribute to an improved labelling of occluded objects, assuming occluded objects show some spatial continuity. The interaction model between neighbouring image sites considers the relative frequency of class transitions, which is different from standard interaction terms such as the contrast-sensitive Potts-Model [2]. In this paper we also propose a new interaction model between two *tCRF* layers, wich is based on directed graph edges - *tCRFd*. For the data-dependent terms we use Gaussian Mixture Models (*GMM*) [16]. Training of *GMM* is frequently done by Expectation Maximization (*EM*), which, due to its iterative nature, is relatively slow and requires all the training data to be held in memory [14]. An alternative method for estimation *GMM* parameters could be the sequential Monte Carlo method, also known as Partcle Filters (*PF*) [7], which are usually used to estimate Bayesian models. Despite the fact that the *PF* have sequential nature, they are still based on the simulation model and therefore are very memory demanding. In order to reduce the memory consumption and to speed up training, we propose a new sequential learning scheme which is considerably faster than *EM*. Our method is demonstrated on the task of correctly labelling urban scenes containing crossroads, one of the major problems in road extraction [15], with the main goal of correctly predicting the class labels of image sites corresponding to the road surface.

2 Conditional Random Fields (CRF)

We assume an image $\mathbf{y}$ to consist of M image sites (pixels or segments) $i \in \mathbb{S}$ with observed data $\mathbf{y}_i$, i.e., $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M)^T$, where $\mathbb{S}$ is the set of all sites. With each site i we associate a class label x_i from a given set of classes $\mathbb{C}$. Collecting the labels x_i in a vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)^T$, we can formulate the classification problem as finding the label configuration $\hat{\mathbf{x}}$ that maximises the posterior probability of the labels given the observations, $p(\mathbf{x}|\mathbf{y})$. A CRF is a model of $p(\mathbf{x} \mid \mathbf{y})$ with an associated graph whose nodes are linked to the image sites and whose edges model interactions between neighbouring sites. Restricting ourselves to a pairwise interactions, $p(\mathbf{x}|\mathbf{y})$ can be modelled by [12]:

$$p(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z} \prod_{i \in \mathcal{S}} \varphi_i(x_i, \mathbf{y}) \prod_{j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j, \mathbf{y}). \tag{1}$$

In Eq. 1, $\varphi_i(x_i, \mathbf{y})$ are the *association potentials* linking the observations to the class label at site i, $\psi_{ij}(x_i, x_j, \mathbf{y})$ are the *interaction potentials* modelling the

dependencies between the class labels at two neighbouring sites i and j and the data $\mathbf{y}$, $\mathcal{N}_i$ is the set of neighbours of site i, and Z is a normalizing constant.

3 Two-Level Conditional Random Fields (tCRF)

In order to classify partially occluded regions we distinguish objects corresponding to the *base level*, i.e. the most distant objects that cannot occlude other objects but could be occluded, from objects corresponding to the *occlusion level*, i.e. all other objects. In a *two-level CRF*, two class labels $x_i^b \in \mathbb{C}^b$ and $x_i^o \in \mathbb{C}^o$ are determined for each image site i. They correspond to the base and occlusion levels, respectively; $\mathbb{C}^b$ and $\mathbb{C}^o$ are the corresponding sets of class labels with $\mathbb{C}^b \bigcap \mathbb{C}^o = \emptyset$. In general, one occlusion level is sufficient for remote sensing imagery. In our application, $\mathbb{C}^b$ consists of classes such as *road* or *building*, whereas $\mathbb{C}^o$ includes classes such as *car* and *tree*. $\mathbb{C}^o$ includes a special class *void* $\in \mathbb{C}^o$ to model situations where the base level is not occluded. We model the posterior probability $p(\mathbf{x}^b, \mathbf{x}^o|\mathbf{y})$ directly, expanding the model in Eq. 1:

$$p(\mathbf{x}^b, \mathbf{x}^o|\mathbf{y}) = \frac{1}{Z} \prod_{i \in \mathbb{S}} \xi_i(x_i^b, x_i^o) \prod_{l \in \{o,b\}} \varphi_i^l(x_i^l, \mathbf{y}) \prod_{j \in \mathcal{N}_i} \psi_{ij}^l(x_i^l, x_j^l, \mathbf{y}). \tag{2}$$

The *association potentials* $\varphi_i^l, l \in \{o, b\}$ link the data $\mathbf{y}$ with the class labels x_i^l of image site i. They are related to the probability of a site i to take labels x_i^l given all image data $\mathbf{y}$ and ignoring the effects of other sites in the image. The *within-level interaction potentials* ψ_{ij}^l, model the dependencies between the data $\mathbf{y}$ and the labels at two neighbouring sites i and j at each level. They are related to the probability of how likely the two sites at level l are to take the labels x_i^l and x_j^l given the image data $\mathbf{y}$. Finally, an *inter-level interaction potential* $\xi(x_i^b, x_i^o)$ is defined to model the dependencies between labels from different levels, x_i^b and x_i^o. It expresses how likely an object from the base level with class label x_i^b could be occluded by an object from the occlusion level with class label x_i^o, ignoring the data $\mathbf{y}$. Fig. 1 shows the structure of our *tCRFd* model. Two levels are split mainly to increase the accuracy of the labelling of occluded regions, where the association potentials cannot provide the base level nodes with reliable information because the corresponding data are not observable.

Training the parameters of the potentials in Eq. 2 requires fully labelled training images. The classification of new images is carried out by maximizing the posterior probability in Eq. 2. Our definitions of the potentials and the techniques used for training and inference are described in the subsequent sections.

3.1 Potential Functions

Association Potential: Omitting the superscript indicating the level of the model, the association potentials $\varphi_i(x_i, \mathbf{y})$ are related to the probability of a label x_i taking a value c given the data $\mathbf{y}$ by $\varphi_i(x_i, \mathbf{y}) \propto p(x_i = c|\mathbf{f}_i(\mathbf{y}))$ [12], where the image data are represented by site-wise feature vectors $\mathbf{f}_i(\mathbf{y})$ that may depend on all the data $\mathbf{y}$. The definition of $\mathbf{f}_i(\mathbf{y})$ may vary with the dataset. We use a *GMM* for $p(x_i = c \mid \mathbf{f}_i(\mathbf{y}))$ [16]:

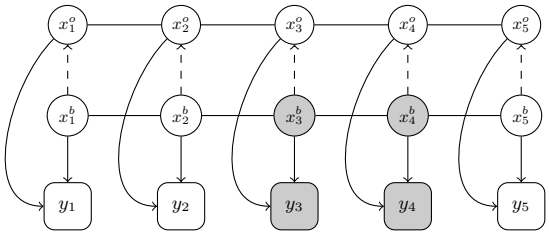

Fig. 1. Structure of the tCRF model. The second dimension and additional links between data and labels are omitted for simplicity. Squares: observations; circles: labels. The dark nodes correspond to a region with occlusion.

$$p(x_i = c \mid \mathbf{f}_i(\mathbf{y})) = \sum_{k=1}^{N_c} \pi_{ck} \cdot \mathcal{N}(\mathbf{f}_i(\mathbf{y}), \mu_{ck}, \mathbf{\Sigma}_{ck}). \tag{3}$$

In Eq. 3, $\mathcal{N}(\mathbf{f}_i(\mathbf{y}), \mu_{ck}, \mathbf{\Sigma}_{ck})$ is the probability density function of a Gaussian with expectation μ_{ck} and covariance matrix $\mathbf{\Sigma}_{ck}$, and π_{ck} are the mixture components measuring the contribution of cluster k to the joint probability density of class c. For each class c there are N_k sets of parameters π_{ck}, μ_{ck}, $\mathbf{\Sigma}_{ck}$. This applies to the models both for the base and for the occlusion levels, i.e. $\varphi_i^b(x_i^b, \mathbf{y})$ and $\varphi_i^o(x_i^o, \mathbf{y})$. The parameters for each class are determined from training data independently from each other, using a sequential learning approach explained in Section 3.3. In our experiments, we compare the *tCRF* model based on the GMM association potential with a model that uses a *naive Bayes* model with $p(x_i = c|\mathbf{f}_i(\mathbf{y})) = \prod_k p(f_i^k|x_i = c)$, where f_i^k is the k^{th} element of $\mathbf{f}_i(\mathbf{y})$ and the probabilities $p(f_i^k|p(x_i = c)$ are derived from the histograms of the feature f_i^k [1].

Within-Level Interaction Potential: This potential describes how likely the pair of neighbouring sites i and j is to take the labels $(x_i, x_j) = (c, c')$ given the data: $\psi_{ij}(x_i, x_j, \mathbf{y}) \propto p(x_i = c, x_j = c'|\mathbf{y})$ [12]. We generate a 2D histogram $h'_\psi(x_i,\ x_j)$ of the co-occurrence of labels at neighbouring sites from the training data; $h'_\psi(x_i = c, x_j = c')$ is the number of occurrences of the classes $(c,\ c')$ at neighbouring sites i and j. We scale the rows of $h'_\psi(x_i,\ x_j)$ so that the largest value in a row will be one to avoid a bias for classes covering a large area in the training data, which results in a matrix $h_\psi(x_i,\ x_j)$. We obtain $\psi_{ij}(x_i, x_j, \mathbf{y})$ by applying a penalization depending on the Euclidean distance $d_{ij} = \|\mathbf{f}_i(\mathbf{y}) - \mathbf{f}_j(\mathbf{y})\|$ of the feature vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ to the diagonal of $h_\psi(x_i,\ x_j)$:

$$\psi_{ij}(x_i, x_j, \mathbf{y}) \equiv \psi_{ij}(x_i, x_j, d_{ij}) = \begin{cases} \frac{\lambda}{\sqrt{\lambda^2 + d_{ij}^2}} \cdot h_\psi(x_i,\ x_j) & \text{if } x_i = x_j \\ h_\psi(x_i,\ x_j) & \text{otherwise} \end{cases} \tag{4}$$

In Eq. 4, λ determines the relative weight of the within-level interaction potential compared to the association potential. As the largest entries of $h_\psi(x_i,\ x_j)$ are usually found in the diagonals, a model without the data-dependent term in Eq. 4 would favour identical class labels at neighbouring image sites and, thus, result in a smoothed label image. This will still be the case if the feature vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ are identical. However, large differences between the features will reduce the

impact of this smoothness assumption and make a class change between neighbouring image sites more likely. This model differs from the contrast-sensitive Potts model [2] by the use of the normalised histograms $h_\psi(x_i,\ x_j)$ in Eq. 4. As a consequence, the likelyhood of a class transition depends on the frequency with which it occurs in the training data. Again, the training of the models for the base and the occlusion levels, $\psi^b_{ij}(x^b_i, x^b_j, \mathbf{y})$ and $\psi^o_{ij}(x^o_i, x^o_j, \mathbf{y})$, respectively, are carried out independently from each other using fully labelled training data.

Inter-Level Interaction Potential: This potential describes how likely x^b_i is to take the value $c \in \mathbb{C}^b$ given that the label x^o_i from the occlusion level takes the value $c' \in \mathbb{C}^o$: $\xi_i(x^b_i, x^o_i) = p(x^b_i = c | x^o_i = c')$. We generate a 2D histogram $h'_\xi(x^b_i, x^o_i)$ of the co-occurrence of labels at different layers and the same image site from the training data; $h'_\xi(c, c')$ is the number of image sites in the training data with $x^b_i = c$ and $x^o_i = c'$. The rows of $h'_\xi(x^b_i, x^o_i)$ are scaled so that the largest value in a row will be one, resulting in a matrix $h_\xi(x^b_i, x^o_i)$ that is the basis for the potential $\xi_i(x^b_i, x^o_i)$. Scaling is necessary to avoid a bias for classes covering a large area in the training data. In our experiments we will compare two different models for handling the inter-level interaction potential. Our model *tCRFd* corresponds to Fig. 1, where the edges connecting the two levels are directed, whereas in the state-of-the-art *tCRFu* model, the edges connecting the two levels are undirected. This difference only affects the inference (Section 3.2).

3.2 Training and Inference

Exact methods for training and inference of a CRF are computationally intractable [12, 21]. Thus, approximate solutions have to be used. We determine the parameters of all potentials separately. The interaction potentials are derived from histograms of the co-occurrence of classes at neighbouring image sites in the way described in Section 3.1. The parameter λ in Eq. 4 is set to $\lambda = 4$, which was determined empirically; it could also be determined by a procedure such as cross validation [20]. The training of the GMM is described in Section 3.3. For inference we use Loopy Belief Propagation, a standard message passing technique for probability propagation in graphs with cycles [21]. In the model *tCRFu*, messages are sent from the base to the occlusion level and vice versa; in *tCRFd*, messages will only be sent from the occlusion level to the base level.

3.3 Sequential Gaussian Mixture Model Training

EM requires the simultaneous storage and processing of all the training samples and the prior definition of the number N_k of Gaussians in the mixture model [14]. In order to overcome these problems we propose a sequential training method for estimating the GMM parameters (cf. Algorithm 1). It requires two parameters, a threshold d_θ defining the minimum distance between Gaussians, and the maximum number $\mathbb{G}_{max}$ of Gaussians in the mixture model. We consider each training sample as an evidence for parameters μ_{ck} and $\mathbf{\Sigma}_{ck}$ of one of the Gaussians in Eq. 3. The samples are processed sequentially in the order in which they are collected. For each new sample we check whether it belongs to an existing Gaussian component k by evaluating the Euclidian distances d_k between

the new sample and the centres μ_{ck} of the existing components. If the smallest distance d_{min} is shorter than d_θ, the sample is assigned to the component k_{min} corresponding to d_{min}, and the parameters of that component are updated. This will affect the centre μ_{ck} of that component, and consequently we check whether we can merge it with any of the others, this time by comparing the Euclidean distances of the class centres to the threshold d_θ. This is done to avoid having too many components. If the training sample does not fit to any existing component (which is, of course, the case for the first training sample to be processed), we generate a new Gaussian component and initialise its centre μ_{ck} by that sample. However, this is only done if the number of Gaussian components is lower than the limit $\mathbb{G}_{max}$, otherwise we discard the training sample. This method is fast because no iterations are required, and it does not require much memory due to its sequential nature. Moreover, we do not need to define the strict number of Gaussians in the GMM, but this number is adjusted to the training data.

Algorithm 1: Sequential GMM training

Data: distance threshold d_θ; max. number of Gaussians $\mathbb{G}_{max}$; *sample points*;
Result: $GaussianMixture$

```
while sample points do
    p ← GetNextPoint();
    if GaussianMixture.N = 0 then
        N ← new Gaussian();
        N.AddPoint(p);
        GaussMixture.Append(N);
    else
        for N_k ∈ GaussianMixture do
            d_k = distance(p, N_k.μ);
        (d_min, k_min) ← MIN(d_k);
        if (d_min > d_θ) AND (GaussianMixture.N < G_max) then
            N ← new Gaussian();
            N.AddPoint(p);
            GaussMixture.Append(N);
        else
            N_{k_min}.AddPoint(p);
        for N_k, N_m ∈ GaussianMixture, k ≠ m do
            d = distance(N_k.μ, N_m.μ);
            if d < d_θ then
                N_k.MergeWith(N_m);
                GaussianMixture.Erase(N_m);
```

4 Features

Our experiments are based on a *colour infrared* (CIR) image (orthophoto) and a *digital surface model* (DSM) image, where grayvalues represent the height of

the earth's surface, including all objects on it. Having a DSM, it is possible to estimate the *digital terrain model* (DTM), which, in contrast to a DSM, represents the bare ground surface without any objects like plants and buildings. We do this estimation by applying to the DSM a morphological opening filter with a structural element size corresponding to the size of the largest off-terrain structure in the scene, followed by a median filter with the same kernel size. Both CIR and DSM images are defined on the same grid. From these input data, we derive the site-wise feature vectors $\mathbf{f}_i(\mathbf{y})$, consisting of 18 features. For numerical reasons, features are scaled and quantized by 8 bit. We use patches of 5×5 pixels as image sites for calculating $\mathbf{f}_i(\mathbf{y})$.

In this paper we use the following features: the *normalized difference vegetation index*, derived from the near infrared and the red band of the CIR image; the *saturation* component in LHS colour space; the *intensity*, calculated as the average of the blue and green channels. These 3 features are derived at 3 different scales: for the individual sites and as the average in a local neighbourhood of 10×10 and 100×100 pixels. Next we determine the *variances of intensity*, *saturation* and the *gradient* determined in a local neighbourhood of 13×13 pixels of each site. Road pixels are usually found in a certain distance either from road edges or road markings. The distance of an image site to its nearest edge pixel is used as the next feature. We also use *histograms of oriented gradients* (HOG) features [5], calculated for cells of 7×7 pixels and using blocks of 2×2 cells for normalization. Each histogram consists of 9 orientation bins. The gradient directions are determined relative to the main direction of the entire scene, supposed to correspond to the direction of one of the intersecting roads. We extract three features from the HOG descriptor, namely the value corresponding to the main direction and the values at its two neighbouring bins. Finally, we use the height difference between the DSM and the DTM as a feature, corresponding to the relative elevation of objects above ground. The last feature is the gradient strength of the DSM.

5 Evaluation

To evaluate our model we selected 90 crossroads from the Vaihingen data set[1]. For each crossroad, a CIR and a DSM were available, each covering an area of 80×80 m^2 with a ground sampling distance of 8 cm. The DSM and the orthophoto were generated from multiple aerial CIR images using semi-global matching [9]. Given our definition of the image sites, each graphical model consisted of 200×200 nodes. The neighbourhood $\mathcal{N}_i$ of an image site i is chosen to consist of the direct neighbours of i in the data grid. We defined 6 classes, namely *asphalt* (*asp.*), *building* (*bld.*), *tree*, *grass* (*gr.*), *agricultural* (*agr.*) and *car*, so that $\mathbb{C}^b = \{asp., bld., gr., agr.\}$ and $\mathbb{C}^o = \{tree, car, void\}$. The two-level reference was generated by manually labeling the orthophotos using these classes and assumptions about the continuity of objects such as road edges in occluded areas to define the reference of the base level. Other existing benchmark datasets

[1] Provided by German Society for Photogrammetry, Rem. Sensing and Geoinf. [4].

could not be used, because they are supplied with a one-layer reference only. For the evaluation we used cross validation. In each test run, 89 images were used for training, and the remaining one for testing. This was repeated so that each image was used as a test image once. The results were compared with the reference; we report the completeness / correctness of the results per class and the overall accuracy [17]. Our CRF-classification is based on the DGM C++ library [6].

We carried out 3 sets of experiments. In the 1^{st} set, we used the *naive Bayes* model for the association potentials (*Bayes*), whereas in the 2^{nd} set we used GMM training based on the OpenCV implementation of EM [3] ($emGMM$). Finally, in the 3^{rd} set we evaluate our sequential GMM model ($seqGMM$). Each set included three experiments: in experiment *CRF*, each layer was processed independently, thus the inter-level interaction potentials were not considered; Experiments *tCRFu* and *tCRFd* used $tCRF$ model with the inter-level interaction potentials represented by undirected and directed edges, respectively (cf. Section 3.1). The completeness and the correctness of the results achieved in these experiments are shown in Tab. 1. Fig. 2 shows the results for three crossroads.

	Bayes						*emGMM*						*seqGMM*					
	CRF		*tCRFu*		*tCRFd*		*CRF*		*tCRFu*		*tCRFd*		*CRF*		*tCRFu*		*tCRFd*	
	Cm.	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*	*Cm.*	*Cr.*
asp.	83.7	77.1	85.7	73.9	82.8	81.2	93.8	58.0	86.6	81.6	83.3	86.6	93.7	58.2	86.5	81.7	83.8	86.5
bld.	65.7	92.2	63.8	93.3	75.4	88.3	72.2	92.0	73.6	91.6	81.0	86.7	72.0	92.2	73.6	91.7	81.0	86.8
gr.	51.6	75.9	88.9	77.2	88.3	81.5	58.5	81.0	91.2	78.8	90.1	83.4	58.6	81.0	91.5	78.7	90.2	83.9
agr.	36.2	96.9	38.4	95.8	68.3	81.5	47.5	97.4	47.9	96.9	81.0	88.5	47.5	97.9	47.8	97.2	80.9	88.5
OA$_b$	**69.1**		**81.5**		**82.3**		**71.0**		**82.3**		**85.7**		**71.2**		**82.4**		**85.6**	
void	85.8	94.3	88.3	93.5	85.8	94.3	90.2	91.9	90.7	91.1	90.0	92.5	90.4	92.0	90.6	91.2	90.4	92.0
tree	82.9	58.5	79.0	61.9	82.9	58.5	71.6	64.2	64,8	70.5	71.7	64.5	71.8	64.3	64,7	70.7	71.8	64.3
car	0.5	17.8	0.0	17.7	0.5	17.8	1.2	40.5	46.7	13.4	1.3	40.9	1.3	40.7	46.5	13.3	1.3	40.7
OA$_o$	**84.5**		**85.8**		**84.5**		**86.1**		**87.2**		**86.2**		**86.0**		**87.3**		**86.0**	
t_t	9.7 *sec*						546.0 *sec*						89.8 *sec*					
t_c	6.4 *sec*						64.0 *sec*						12.5 *sec*					
RAM	1.2 *MB*						2.44 *GB*						1.5 *MB*					

Table 1. Completeness ($Cm.$), Correctness ($Cr.$), overall accuracy (OA) [%] and timings for Intel® Core™ i7 CPU 950 with 3.07 GHz required for training (t_t) and classification (t_c).

In the $seqGMM$:CRF experiment, the overall accuracy of the classification was 71.2% for the base level and 86.0% for the occlusion level. Considering the inter-level interactions in the $seqGMM$:$tCRFu$ and $seqGMM$:$tCRFd$ experiments increased the overall accuracy for the base level by 11% - 14%, with a slight advantage for the model based on directed edges ($seqGMM$:$tCRFd$ with OA_b=85.6%). This can be attributed by more accurate classification in the occlusion areas (cf. Fig. 2, particularly the areas where roads are partially occluded by trees). For the occlusion level, the overall accuracies of the $seqGMM$:$tCRFu$ and $seqGMM$:$tCRFd$ experiments were 87.3% and 86.0%, respectively; in this case, there is hardly any improvement over the variant not considering the inter-level interactions $seqGMM$:CRF, and the model $seqGMM$:$tCRFu$ performed slightly better than the others. In all experiments, the results based on $seqGMM$ are very similar to those achieved for the $emGMM$ and in the same time are better than those achieved for the *Bayes* model, having similar behaviour. As far as completeness and correctness are concerned, the major improvement is an

increased correctness for *asp.* and an improved completeness for *gr.*. The class *agr.*, corresponding to fields, has a rather low completeness in the model $tCRFu$, though a much better one in $tCRFd$. For the occlusion level, we observe the best performance when using $tCRFu$. As we can see from the Tab. 1, it achieved the best classification rate of the *car* class, though both the completeness and correctness of that class are still very low. This may be due to the fact that cars are small compared to the size of an image site (40 cm). Without base level support, they are smoothed out in the occlusion level (cf. Fig. 2).

The computation times for training the $tCRFd$ model on 89 images were 9.7 sec and 89.8 sec for the *Bayes* and *seqGMM* models, respectively; the time for inference was 6.4 sec and 12.5 sec, respectively, per image. The memory consumption was slightly above 1 MB in both cases. For the $emGMM$ experiments the computation times were 546.0 sec for training and 64.0 sec per image for inference, with a memory consumption of 2.44 GB. So $seqGMM$ is much closer to the *Bayes* in terms of calculation time and memory requirements, while being close to $emGMM$ in terms of classification accuracy.

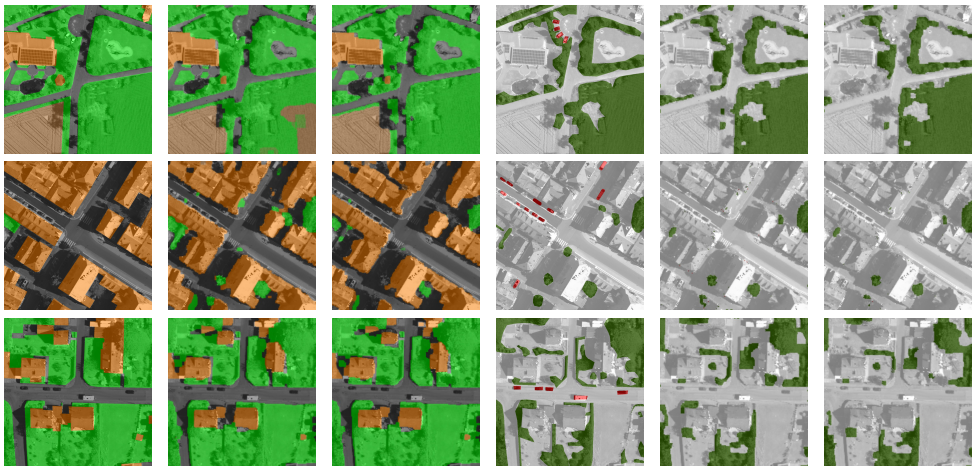

Fig. 2. Three example crossroads. 1^{st} col.: reference; 2^{nd} col.: CRF; 3^{rd} col.: tCRFu. Occlusion level: 4^{th} col.: reference; 5^{th} col.: CRF; 6^{th} col.: tCRFu.

6 Conclusion

We have presented a sequential approach for GMM training, which supports a $tCRF$ model for considering occlusions in classification. Due to the two-level structure and incorporation of directed edges our model is capable of improving the accuracy of classification for partially occluded objects. Our sequential approach is more than 50 times faster and needs far less memory than classical EM. The method was evaluated on a set of airborne images and showed a considerable improvement of the overall accuracy in comparison to the CRF and *Bayes* approaches. In the future we want to extend the model to an n-level architecture, which will require the removal of the restriction $\mathbb{C}^b \bigcap \mathbb{C}^o = \emptyset$.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York, NY (USA), 1^{st} edn. (2006)
2. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: Proc. ICCV. vol. I, pp. 105–112 (2001)
3. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
4. Cramer, M.: The DGPF test on digital aerial camera evaluation - overview and test design. Photogrammetrie-Fernerkundung-Geoinformation 2(2010), 73–82 (2010)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR. pp. 886–893 (2005)
6. DGM: Direct graphical models library. http://research.project-10.de/dgm/ (2013)
7. Fearnhead, P.: Particle filters for mixture models with an unknown number of components. Statistics and Computing 14(1), 11–21 (2004)
8. Hinz, S., Baumgartner, A.: Automatic extraction of urban road networks from multi-view aerial imagery. ISPRS J. Photogramm. & Rem. Sens. 58, 83–98 (2003)
9. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. 30(2), 328–341 (2008)
10. Kosov, S., Kohli, P., Rottensteiner, F., Heipke, C.: A two-layer conditional random field for the classification of partially occluded objects. http://arxiv.org/abs/1307.3043 (2013), arXiv:1307.3043 [cs.CV]
11. Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. In: Proc. ICCV. pp. 1284–1291 (2005)
12. Kumar, S., Hebert, M.: Discriminative Random Fields. Int. J. Comput. Vis. 68(2), 179–201 (2006), http://www.springerlink.com/index/10.1007/s11263-006-7007-9
13. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. Int. J. Comput. Vis. 77, 259–289 (2008)
14. McLachlan, G., Krishnan, T.: The EM algorithm and extensions. Wiley series in probability and statistics, Wiley, Hoboken, NJ, 2^{nd} edn. (2008)
15. Ravanbakhsh, M., Heipke, C., Pakzad, K.: Road junction extraction from high resolution aerial imagery. Photogrammetric Record 23, 405–423 (2008)
16. Reynolds, D.A.: Gaussian mixture models. In: Encyclopedia of Biometrics, pp. 659–663. Springer US (2009)
17. Rutzinger, M., Rottensteiner, F., Pfeifer, N.: A comparison of evaluation techniques for building extraction from airborne laser scanning. JSTARS 2(1), 11–20 (2009)
18. Schindler, K.: An overview and comparison of smooth labeling methods for land-cover classification. IEEE-TGARS 50, 4534–4545 (2012)
19. Schnitzspan, P., Fritz, M., Roth, S., Schiele, B.: Discriminative structure learning of hierarchical representations for object detection. In: CVPR. pp. 2238–2245 (2009)
20. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int. J. Comput. Vis. 81, 2–23 (2009)
21. Vishwanathan, S.V.N., Schraudolph, N.N., Schmidt, M.W., Murphy, K.P.: Accelerated training of conditional random fields with stochastic gradient methods. In: Proc. 23^{rd} ICML. pp. 969–976 (2006)
22. Winn, J., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: Proc. CVPR (2006)
23. Yang, Y., Hallman, S., Ramanan, D., Fowlkes, C.: Layered object detection for multi-class segmentation. CVPR (2010)