Sequential Gaussian Mixture Models for Two-Level Conditional Random Fields

Sergey Kosov, Franz Rottensteiner, Christian Heipke

Leibniz Universität Hannover

Institute of Photogrammetry and GeoInformation
Hannover, Germany

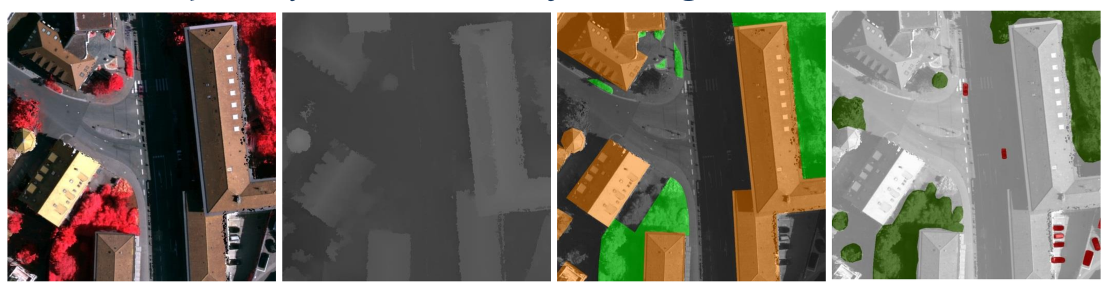
This work addresses the problem of efficient classification of partially occluded objects. For this purpose we propose a novel Gaussian Mixture Models based on a sequential training procedure, in combination with multi-level Conditional Random Fields framework.

1 First, we extend the state-of-the-art Conditional Random Fields (CRF) framework to the two-layer architecture, which allows us to consider our problem not only in 2D space, but also in third dimension. This extension gives us a tool for proper classification of partially occluded regions. 2 The main disadvantage of the two-layer architecture is that it tends to make CRF slower, especially as the number of nodes is increased by adding additional layers. For estimating the node potentials the Gaussian Mixture Models (GMM) are used. The classical approach for the GMM training is the Expectation Maximization (EM) algorithm. In our work we present the sequential GMM training algorithm, which significantly improves the training and classification speed, while keeping near the same classification accuracy. 3 GMM approximate the distribution of training sample points in the multi-dimensional feature space. In this work we use sample points, consisting of 18 features, which are derived from the original data and describe it. We make use of radiometric- and depth-map based features, as well as multi-scale features. 4 The experiential results show the advantage of the novel sequential GMM training algorithm over the classical EM algorithm.

Setup and Algorithm

- Dataset of color-infrared airborne images of urban area, which includes:
  - orthophotos with the resolution of 1000x1000 pixels and the ground sampling distance of 8 *cm*
  - digital surface model (DSM) with the same resolution (raster terrain elevation map)
  - one graph node corresponds to an image patch of size 5x5 pixels
- Classification into 6 classes: *asphalt, building, tree, grass, agricultural, car* and *void*. (Gaussian mixture models, supported by conditional random fields)

An examples of dataset entry with groundtruth:



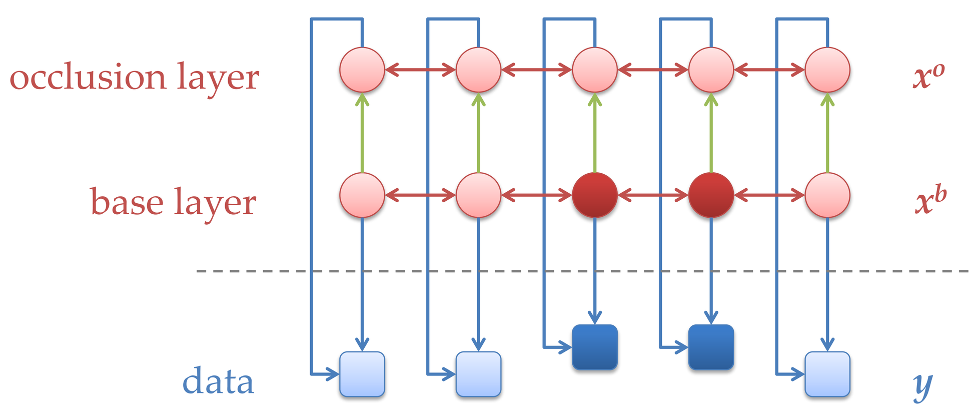
1 Two-Layer CRF Model

- Assigns *two* labels $\boldsymbol{x}^b$ and $\boldsymbol{x}^o$ to each data site:
  - *base layer* labels $\boldsymbol{x}^b$ are for most distant objects that cannot occlude other objects
  - *occlusion layer* labels $\boldsymbol{x}^o$ are for those objects that may occlude objects from the base layer
- Explicitly models 3D structure of the scene and therefore may deals with occlusions:

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

- Class *void* lets us to model situations where the base level is not occluded

Structure of the tCRF model for 1D case. Squares: observations; circles: labels. The dark nodes correspond to a region with occlusion:



2 Sequential GMM Training

- The classical EM algorithm has iterative nature and requires all the training data to be held in memory
- The EM algorithm also requires the predefined number Gaussian in the mixture and therefore less flexible
- Sequential algorithm for GMM training is memory and CPU time inexpensive
- Number of Gaussians in the mixture is self-adjustable according to the training data
- **Input Data:**
  - distance threshold d_θ;
  - maximal number of Gaussians $\mathbb{G}_{max}$;
  - sample points;
- **Result:**
  - *GaussianMixture*

The sequential GMM training algorithm:

```
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);
```

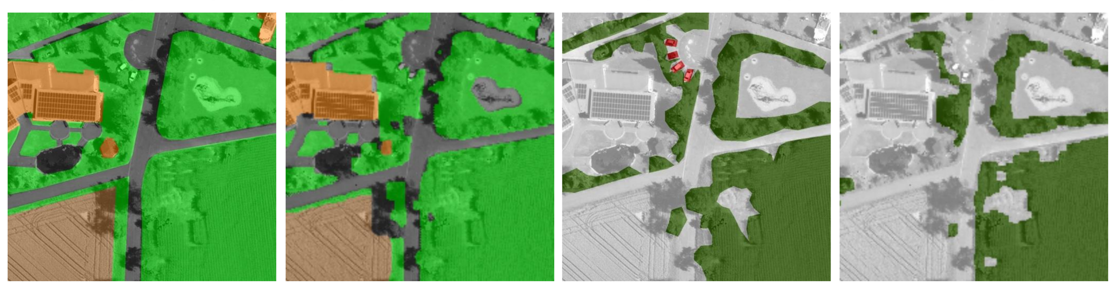
3 Data Features

- Normalized difference vegetation index, image saturation and intensity, derived at 3 different scales: *i.e.* at 1 pixel, 10x10 and 100x100 pixels
- Variances of image intensity, saturation and gradient, in the neighborhood of 13x13 pixels
- Distance transform map of the image gradient
- Histogram of oriented gradients
- DSM - based feature, and its gradient

4 Experimental Results

- Cross Validation was used with training on 89 test images and classification one test image
- Comparison of naïve Bayes training (*Bayes*), state-of-the-art GMM training based on EM algorithm (*emGMM*) and our sequential GMM training (*seqGMM*)

Example of classification with seqGMM for both layers



● *asp.;* ● *bld.;* ● *tree;* ● *grass;* ● *agr.;* ● *car;* ○ *void*

- **Classification Accuracy**
- Results of *emGMM* and *seqGMM* are shown to be very close (difference is less than 1%)

Overall accuracy (OA) of classification for base and occlusion layers

	Bayes			emGMM			seqGMM		
	CRF	tCRF	tCRFd	CRF	tCRF	tCRFd	CRF	tCRF	tCRFd
OA_{base}	69,1%	81,5%	82,3%	71,0%	82,3%	85,7%	71,2%	82,4%	85,6%
OA_{occl}	84,5%	85,8%	84,5%	86,1%	87,2%	86,2%	86,0%	87,3%	86,0%

- **Performance and efficiency**
- Training time improvement is up to 6 times
- Classification time improvement is up to 5 times
- Memory consumption t is $1,6\text{x}10^3$ times less

Comparison of CPU time required for training and cclassification as well as memory consumption

	Bayes	emGMM	seqGMM
$t_{training}$	9,7 sec	546,0 sec	89,8 sec
$t_{classification}$	6,4 sec	64,0 sec	12,5 sec
RAM	1,2 MB	2,44 GB	1,5 MB