NON-ASSOCIATIVE MARKOV NETWORKS FOR 3D POINT CLOUD CLASSIFICATION

Roman Shapovalov, Alexander Velizhev, Olga Barinova

Graphics & Media Lab, Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Room 701, bld. 52, 1 Leninskie Gory, Moscow, 119992 Russia
{shapovalov, avelizhev, obarinova}@graphics.cs.msu.ru

Commission WG III/2

KEY WORDS: LIDAR, Classification, Point Cloud, Random Forest, Markov random fields

ABSTRACT:

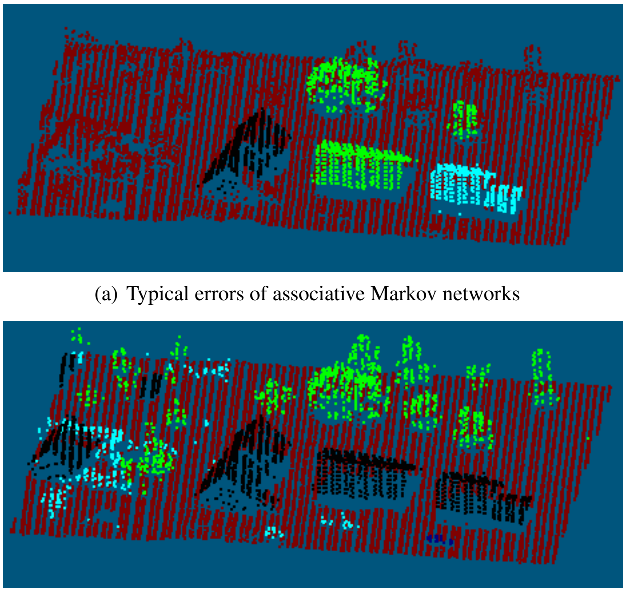
The problem of laser scan analysis gained significant attention within the last decade. The standard approach to point cloud classification utilizes Markov Random Fields (MRF). Usually, a subclass of MRFs, Associative Markov Networks (AMNs), are used. In AMN the pairwise potential function is constant for a pair of different class labels. In some cases this constraint is too rigorous since it does not allow expressing some natural interactions between objects, such as “roof is likely to be above the ground”. In this work we use the general form of pairwise potentials instead. We show how to perform a tractable inference in such network using a message-passing based algorithm. Oversegmentation technique is used to subsample a scan; this helps to eliminate noise, improve efficiency and makes possible to use natural edge features. Experiments on two datasets obtained by an airborne laser scanner show that non-associative Markov networks usage leads to improvement in classification accuracy.

1 INTRODUCTION

In this paper we address the problem of 3D point cloud classification, i.e. we need to assign one of the predefined class labels to each point of a cloud based on local properties of the point and global properties of the cloud. This problem is hard due to noise, occlusions, ambiguity and lack of 3D points corresponding to small objects.

The existing methods for LIDAR data classification could be divided into four groups: filtering, the methods based on unsupervised learning, and on supervised learning with or without using MRFs. The first group (Sithole and Vosselman, 2004) is used for binary classification, e.g. separation of ground and non-ground points, vegetation or buildings extraction. That filters can be applied consequently for extracting different classes of objects. The common thing in these approaches is that filters always operate only on a local neighbourhood of points (Sithole and Vosselman, 2004). This is a drawback because it is often impossible to classify points reliably without consideration of global context. The next group of methods solves the classification problem by performing semantic segmentation (Tovari, 2006), (Chehata et al., 2008).

Supervised learning methods assume that data are separated into training and test parts, and compute different types of features for all points: height-, echo-, eigenvalue- or geometry- based ones (Chehata et al., 2009). The training data features are used to build a classifier which is applied to the test points. SVM (Lodha et al., 2006), AdaBoost (Lodha et al., 2007) or Random Forest (Chehata et al., 2009) classifiers have been used in the literature. (Carlberg et al., 2009) use a cascade of classifiers to distinguish points of each class. The method also uses region-growing segmentation in the 3D domain to enforce a spatial coherency of classification results. On the first stage water and ground classifiers are applied. It limits the region growing space during classification of the other classes. However, in the case of low buildings with neighbouring trees the region-grow algorithm might simply shuffle different classes.



(a) Typical errors of associative Markov networks

(b) Ground truth

Figure 1: AMN is likely to smooth the classification result. Color legend: red-ground, black-building, navy-car, green-tree, cyan-bush.

Since any single point does not give much information, spatial context is widely used for classification. Markov network provides a natural model for interactions between points in a cloud. (Anguelov et al., 2005) were the pioneers in semantic segmentation of 3D point clouds using MRFs, though they based on the related work on semantic segmentation of images.

(Anguelov et al., 2005) used the subclass of MRFs called associative Markov networks (Taskar et al., 2004). Pairwise potentials in AMN generalize the Potts model. In the model neighbouring nodes are encouraged to have the same label: $\log \phi_{ij}(l,l) = \lambda_{ij}^l$, where $\lambda_{ij}^l > 0$, and $\log \phi_{ij}(k,l) = 0$, if $k \neq l$. Since such a potential function is submodular for any pair of class labels, it is reasonable to use alpha-expansion iterative procedure over graph cuts for maximum a posteriori (MAP) inference (Boykov et al., 2001). Unary and pairwise potentials are modelled as a log-linear combination over node and edge features correspondingly: $\log \phi_i(l) = \mathbf{w}_n^l \cdot \mathbf{x}_i$, $\log \phi_{ij}(l, l) = \mathbf{w}_e^l \cdot \mathbf{x}_{ij}$, where $\mathbf{w}_n^l$ and $\mathbf{w}_e^l$ are label-specific model parameters, $\mathbf{x}_i$ and $\mathbf{x}_{ij}$ are node and edge features. For learning those parameters the method proposed by (Taskar et al., 2004) is used. The target is to maximize the margin of energy function value in the true label assignment $\hat{\mathbf{Y}}$ over any different assignment $\mathbf{Y} \neq \hat{\mathbf{Y}}$, similar to how SVM does it. To summarize, an AMN performs intelligent smoothing of classification results. The negative effect could be seen on Figure 1.

As the local features (Anguelov et al., 2005) use spin images (Johnson and Hebert, 1999) of the size 5×10 reduced by a principal component analysis (PCA) to obtain 45 principal components. The important thing they discovered is that the size of a single bin should be approximately equal to the resolution of the scan. Also, the authors used features that are invariant to rotation in XY plane. One of such features is the distribution of points in the neighbourhood, binned to cells of the $3 \times 3 \times 3$ cube with the center in the reference point and aligned with the surface approximated by its neighbourhood via principal component analysis (PCA). Another such feature is the distribution of heights of points below and above the reference point. They had tried different edge features, but concluded that the constant feature for each edge had performed best.

(Triebel et al., 2006) used a similar algorithm. Their main contribution is using adaptive scan resampling in order to speed classification up. They build a *k*d-tree over the points of a scan and then prune the tree. The set of points within each leaf is replaced by a single point. In (Triebel et al., 2007) the algorithm is revisited. The authors noted that conventional AMN is not enough flexible, because the potentials depend linearly on the features. They combined nearest neighbour classifier with AMN. This led to significant performance growth on their dataset.

(Munoz et al., 2008) continued the work of (Anguelov et al., 2005). They tried to learn more accurate pairwise potentials. The authors use non-constant edge features. The feature vector of an edge is a concatenation of the spectral features of the incident points as well as the similarity of the directional features of the points. Spectral features tell if point's neighbourhood looks like a surface, a line, or noise. For a point, the tangent vector and the normal vector are estimated. Directional features are the angles formed by those vectors against the vertical and the horizontal planes, scaled in some special way.

(Munoz et al., 2009b) used radically different model. They show that a conditional random field (CRF) with higher order cliques helps to increase the precision in addition to significant speedup. The authors use k-means clustering in order to retrieve cliques. The cliques are isolated from the other cliques in the CRF formulation. They use the P^n Potts model for clique potentials. For the clique C_i the potential is $\log \phi_{C_i}(\mathbf{Y}_c) = \mathbf{w}_{C_i}^l \cdot \mathbf{x}_c$, if $Y_i = l \ \forall i \in c$, and $\log \phi_{C_i}(\mathbf{Y}_c) = 0$ otherwise, where c is the set of indices of the clique nodes (Kohli et al., 2007). The potentials are learned by margin maximization via subgradient projection method. Clique features $\mathbf{x}_c$ are the same as the node features but they are computed for the points of a clique instead of the points in the support volume. In (Munoz et al., 2009a) they use functional gradient boosting instead of the subgradient method, which leads to dramatic improvement in quality.

(Lu et al., 2009) used a hybrid CRF for ground detection. The CRF contains both discrete and continuous latent variables. They are used to represent the state of a point (ground or non-ground) and the actual ground elevation under the point (which is equal to the observed elevation for ground points). The authors perform inference using the expectation-maximization (EM) algorithm. They treat the estimated terrain map as parameters and ground classification as hidden variables. Although EM is not guaranteed to find the globally optimal solution, it performs well enough in practice.

We used the classic AMN model as a starting point. Although the AMN model gives significant advantage over classification of 3D points independently, it is still too limited. It generally fails to detect both large and small objects due to oversmoothing. Another problem is error propagation, i.e. if some of the points are incorrectly classified, it might lead to incorrect classification of a larger region due to the same smoothing effect. See the right side of the scan on Figure 1 for instance. Moreover, AMN model does not allow expressing some natural interactions between objects, such as “roof is likely to be above the ground”. In contrast to the papers mentioned above we do not restrict ourselves to the AMN model. In this work we use the general form of pairwise potentials instead. We show how to perform a tractable MAP inference in such network using the TRW-S algorithm (Kolmogorov, 2006). Oversegmentation technique is used to subsample the scan; this helps to eliminate noise, improve efficiency and makes possible to use natural edge features. Experiments on two datasets obtained by an airborne laser scanner show that using non-associative Markov networks leads to improving classification accuracy.

The paper is organized as follows. In the following section we briefly describe the MRF model. In Section 3 we provide a detailed description of the suggested technique. In Section 4 we report on experimental results and then conclude the paper.

2 MARKOV RANDOM FIELDS

Markov random field (Markov network) is a popular graphical model for 3D point cloud segmentation. The segmentation task is formulated in terms of MAP estimation in Markov networks, which is also referred to as energy minimization. In the problem of point cloud classification nodes of a graph correspond to the points of a scan, and minimum of energy corresponds to the best possible class labels assignment to the 3D points. The nodes corresponding to neighbouring points are connected with edges. Consider a scan $\mathbf{I}$ which contains N points. We need to come up with a labeling $\mathbf{Y}$, for each of the 3D points one of K predefined labels should be assigned: $\mathbf{Y} = \{Y_1, \ldots, Y_N\}, Y_i \in \{1, \ldots, K\}$. In pairwise Markov network, for each node and for each edge a non-negative potential function is given as a function of possible assignments: unary potentials $\phi_i(Y_i|\mathbf{I})$ for nodes and pairwise potentials $\phi_{ij}(Y_i, Y_j|\mathbf{I})$ for edges. To emphasize the fact that potential functions are conditioned by features of the scan, the model is also referred to as conditional random field (CRF). The energy is formulated as follows:

$$E(\mathbf{Y}|\mathbf{I}) = -\sum_{i=1}^{N} \log \phi_i(Y_i|\mathbf{I}) \ - \sum_{(i,j)\in E} \log \phi_{ij}(Y_i, Y_j|\mathbf{I}) \quad (1)$$

The problem of minimizing energy of this form is NP-hard whenever $K > 2$ (Kolmogorov and Zabih, 2004), but there exist a number of methods for finding an approximate minimum. The most used of them are graphcut-based methods (Boykov et al., 2001), which are known to be both effective and efficient. However, they are inapplicable when pairwise potentials do not satisfy submodular restrictions. The general problem could be solved approximately by message passing techniques like loopy belief propagation (LBP) (Yedidia et al., 2000) and tree-reweighted message passing (TRW). LBP is likely to fall into endless loop, while TRW is proven to converge to some fixed point, which is often close to the optimum. In practice, TRW also shows better performance than LBP.

Sequential variation of the algorithm (TRW-S) was developed by Vladimir Kolmogorov (Kolmogorov, 2006). It is proven to find the global minimum of the concave lower bound on the energy function. The graph is spanned by a set of so-called monotonic chains, i.e. chains, that have their nodes ordered according to some global order on the nodes of the graph. Each node of the graph should be contained by at least one chain. For each chain there exists its own parameter vector. The tightest lower bound on the energy is formulated as a concave function of those vectors. For further information please see (Kolmogorov, 2006).

3 PROPOSED METHOD

The workflows for learning and classification stages are similar, so we describe them in parallel. First, a spatial index is built over test/train set, which is also considered as segmentation. Then a graph over segment medoids is built. After that, features for unary and pairwise potentials are computed. For unary potentials computation, Random Forest classifier is used. To train it the subsampled set of point features from the train set is used. During the classification stage all segment medoids are classified by the trained Random Forest classifier. For pairwise potentials computation Naïve Bayes classifier is used. It is trained on the features extracted from the train set graph. When unary and pairwise potentials are computed, MRF inference returns the final label assignment. These stages are described in the appropriate subsections.

3.1 Spatial Index and Segmentation

Spatial index allows rapid performing spatial queries (like k nearest neighbours retrieval). For this purpose we use modified R-Tree data structure (Guttman, 1984). Instead of the split that attempts to minimize the sum of volumes of the resulting bounding boxes, we used k-means clustering as a split algorithm. Also, we implemented a tailored strategy for inserting points. A point is to be inserted to the leaf with the closest centroid. Since search of the best leaf is computationally intensive, an heuristic for search space reduction is used.[1] These modifications led to the compact clusters of points within each leaf, so we consider the leaf structure as a form of oversegmentation. See Figure 3(a) for example of segmentation result. On the classification stage for each segment we classify only its medoid and then spread the results to the other points of the segment. This technique helps to cope with acquisition and labelling noise, but fails to find objects of sub-segment size. The possible loss caused by segmentation is analyzed in Section 4.1. However, this form of subsampling leads to dramatic classification speedup. It is tractable to classify millions of points using a single model.

3.2 Graph Construction

Both in training and classification stage we need to build a graph over medoids of the segments in order to perform MRF inference. The common approach to building graph is to add all the edges that connect points to their k nearest neighbours. The typical value of k for AMN varies from 3 to 5 (Munoz et al., 2008). Since our pairwise potentials are not "attractive", there is nothing bad in adding more edges to a graph. Thus, we add all 5-connected edges.

Aerial scans have its specific. They are scanned from the top, hence there are a lot of discontinuities instead of the vertical surfaces such as walls and fences. For example, the roof of a building or the crown of a tree could be isolated from the ground in the graph. Since the edges between such segments are extremely useful, we use the following workaround: in addition to the 5NN edges we add to the graph all the 5-connected edges in the cloud projection to a horizontal plane.

3.3 Features and Potentials

3.3.1 Unary Potentials. We use the output of the Random Forest algorithm (Breiman, 2001) as unary potentials. Since Random Forest consists of multiple trees (we use 50)[2], it is possible to assign the value of the potential function of each class label proportionally to the number of trees voted for the label. On the Figure 3(b) one can see a typical probabilistic output of the multiclass Random Forest, which is the unary potential function as well.

To train Random Forest it is not necessary to use all the train set points. It is reasonable to do random subsampling. We subsample the points of the different classes proportionally to the class frequency, so there is nearly equal number of points of each class in the train set. According to (Elkan, 2001), a classifier trained on the balanced data yields an unbiased classification result. The following local features are used for classification:

- spectral and directional features as described in (Munoz et al., 2008), totally 7 real-valued features;
- spin images of size 9×18 (Johnson and Hebert, 1999), with simple dimensionality reduction applied, totally 27 features;
- angular spin images of size 9×18 (Endres et al., 2009), also with reduced dimensionality, totally 27 features;
- distribution of heights of the points in infinite vertical cylinder around the point (approximated with a 5-bin histogram), minimum point height in the cylinder and difference of the point height with the min height, totally 7 features.

In total, we use 68 real-valued Random Forest features. Experiments showed that dropping any group of features or reducing spin image size leads to performance degradation, so we consider this set irredundant.

3.3.2 Pairwise Potentials. Pairwise potentials used in AMNs have the following drawback: they are always null for different class labels ($\log \phi_{ij}(k, l) = 0$, if $k \neq l$). So, it is impossible to express any kind of inter-class relations (like "tree could not be lower than ground") within such model, though it appears to be extremely useful. We do not restrict ourselves with such constraints. Using the graph built on the medoids of train/test scan segments, we compute three edge features:

- cosine of the angle between approximated normals in the medoids: $\mathbf{n}_p \cdot \mathbf{n}_q / (||\mathbf{n}_p|| ||\mathbf{n}_q||)$;
- difference in altitudes of the medoids $\mathbf{p}$ and $\mathbf{q}$ normalized by the distance between them: $(p_z - q_z)/||\mathbf{p} - \mathbf{q}||$;
- distance between medoids $||\mathbf{p} - \mathbf{q}||$.

Please note that the last two features are statistically significant only due to oversegmentation. It is not reasonable to use them if MRF is built over points of a scan.

[1] Implementation and details of the algorithm are available on-line: http://graphics.cs.msu.ru/en/science/research/3dpoint/lidark

[2] OpenCV is used: http://sourceforge.net/projects/opencvlibrary/

Let us denote these three features' values for some edge f_1, f_2 and f_3. Using Bayes' theorem it is possible to compute the probability of assignment the labels l_1 and l_2 if all the features are known:

$$P(l_1 l_2 | f_1, f_2, f_3) = \frac{P(f_1|l_1 l_2)P(f_2|l_1 l_2)P(f_3|l_1 l_2)P(l_1 l_2)}{P(f_1, f_2, f_3)} \quad (2)$$

The experiments have shown that the term $P(l_1 l_2)$ is dominating, i.e. if the prior probability for the pair of classes is big, it is unlikely to get any other assignment, even if the features vote for it. So, we ignore the prior term in the formula. In order to estimate the rest four probabilities in the right side of (2) we discretize feature value range and collect the statistics from the edges of labelled train set scans. Thus, we approximate the distributions with histograms. We refer to this technique as Naïve Bayes learning. During classification pairwise potentials are estimated with respect to the edge features: $\phi_{ij}(k, l) = P(kl|f_1, f_2, f_3)$.

Naïve Bayes classifier fits the problem well. It is simple, nonlinear and allows coping with imbalance easily. However, it treats all edges independently. Structured learning techniques like cutting-plane training of SVM (Joachims et al., 2009), which optimize parameters of the potentials jointly, are promising here.

3.4 Inference

Since we use general form of energy function (without submodular restrictions (Kolmogorov and Zabih, 2004)), graph cut based inference techniques are inapplicable. We tried to use loopy belief propagation and TRW-S[3], the latter turned out to perform better, which corroborates the theoretical guarantees behind it. Although loopy BP sometimes yields the same results as TRW-S, it is less stable. LBP fails more often when an MRF contains more edges.

4 EXPERIMENTAL RESULTS

We evaluate our method on two data sets obtained in different regions using airborne laser scanning system ALTM 2050 (Optech Inc.). We refer to them as airborne datasets A and B.[4] We use the following class labels: "ground", "building", "tree", "low vegetation"; for the first scan "car" label is used as well. The data are distributed among the labels in the following proportion: 43.8% for ground, 2.0% for building, 0.3% for car, 53.1% for tree and 0.8% for low-vegetation in the "dataset A" scan. The proportion is similar for the "dataset B" with the exception that it has no cars. It can be seen that there are few cars in the first scan too, but we consider it as a separate class to make the problem more challenging. We split each sample into a train set and a test set of sizes 1.1M and 1.0M points for "dataset A", and 1.5M and 1.2M points for "dataset B", respectively. To train unary potentials we subsample the train sets with respect to class frequency, which results in unbiased train sets. Thus, the number of training vectors for "dataset A" is reduced to 9K. To train the pairwise potentials we segment the train sets, compute the medoid for each segment, build the graph upon them and compute the statistics of its edges as described in Section 3.3.2. In order to train the Naïve Bayes classifier, we approximate all the distributions with histograms. The altitude difference feature and the directional feature both vary in the range $[-1, 1]$. We discretized them into 10 equally-sized bins. The distance feature is discretized into 6 bins, which are scan-specific. They are found empirically to approximate the distributions well. For "dataset A" the bins' borders are $(2.0, 4.0, 6.0, 8.0, 12.0)$, for "dataset B" they are $(2.5, 3.5, 4.5, 6.0, 8.0)$.

4.1 Least Theoretical Error

In this section we investigate the problem of inaccurate detection of object boundaries due to use of preliminary segmentation. In order to estimate how crucial the problem is, we computed the minimum error, which is possible to get by assigning a single label to each segment. For each segment, we assigned the most frequent ground truth label in the segment to all of its points, and then computed the overall error. The results for both data sets are shown on the Figure 2. The smaller error for the "dataset B" scan could be explained by the absence of the "car" label; the class of cars is unlikely to be segmented well since cars are usually small, and they tend to stick to the ground.

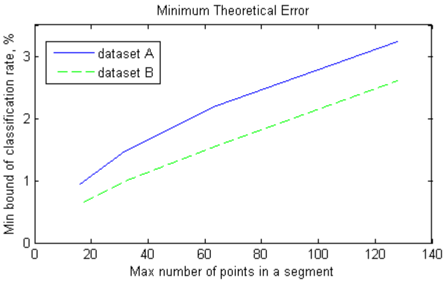


Figure 2: Minimum theoretical error caused by segmentation

The maximum segment size affects the classification error, which is traded-off against the efficiency, because it depends directly on the number of segments. We decided to set the maximum segment size equal to 64 points, which results to the theoretical segmentation error 2.2% for "dataset A" and 1.6% for "dataset B" and about 30K segments for each scan.

4.2 Classification Accuracy

We performed several experiments to compare our method with the existing techniques. First, we perform Random Forest classification without MRF. Second, we test the graph cut algorithm over associative Markov network with constant pairwise potentials ($\log \phi_{ij}(Y_i, Y_j) = [Y_i = Y_j]$), adding to the graph all the edges, which connect nodes to their 3-connected neighbours. It is less than we use for non-associative MRF, because AMN is likely to smoothen results if more edges are used. Third, we show the results for our method.

Tables 1 and 2 demonstrate the confusion matrices for our results, precision and recall for all three featured methods. To compare the results we use the mean f-score metrics. F-score is the harmonic mean of precision and recall. Mean f-score is the average f-score over all classes. We consider it better metrics than overall precision since it treats small classes properly, while they are almost ignored by the overall precision. Mean f-scores for Random Forest, constant AMN and non-associative MRF are 46.7%, 46.8%, **59.3%** and 76.0%, 75.1%, **77.4%** correspondingly for "dataset A" and "dataset B" scans.

Constant AMN smooths the results, removing both classification noise and small details (Figure 3(e)). On the first scan, such smoothing improved the result dramatically, while on the second one it did not help. Non-associative Markov network contained more edges, whose potentials vary depending on the distance between points and other features. This helps to perform more intelligent smoothing (Figure 3(f)). Moreover, the new form of potential function allow potentials that stimulate the relations like "roof is higher than ground".

[3] We used the implementation by Vladimir Kolmogorov: `http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/TRW-S.html`

[4] The labelled sets and their characteristics could be found at `http://graphics.cs.msu.ru/en/science/research/3dpoint/classification`

Our method	ground	building	car	tree	low veg	Recall	RF Recall	AMN recall
ground	395228	696	5606	1227	37232	0.9620	0.9700	0.9640
building	288	17004	9	212	2079	**0.5846**	*0.1959*	*0.0783*
car	1549	77	1196	4	409	0.1608	*0.1082*	**0.2094**
tree	12648	10721	180	491016	17287	**0.9970**	0.9643	*0.9042*
low veg	1124	587	445	49	5553	0.0888	*0.0590*	**0.1243**
Precision	**0.8983**	**0.8679**	0.3697	0.9232	*0.7158*			
RF Precision	0.7889	*0.3761*	**0.4526**	*0.8999*	*0.7379*			
AMN Precision	*0.7476*	*0.0792*	*0.3255*	**0.9879**	**0.8866**			

Table 1: The confusion matrix for "dataset A" scan. All the scan points are included, not only the medoids of the segments

Our method	ground	building	tree	low veg	Recall	RF Recall	AMN recall
ground	958243	395	556	18593	0.9800	*0.9652*	**0.9895**
building	722	43948	3835	6027	*0.8059*	0.8210	**0.8591**
tree	3655	3560	99225	4556	0.8940	**0.9305**	*0.8646*
low veg	5158	1816	364	9607	**0.5670**	0.5396	*0.1822*
Precision	**0.9901**	0.8839	0.9543	0.2477			
RF Precision	0.9902	*0.8101*	*0.9295*	*0.2246*			
AMN Precision	*0.9686*	**0.9301**	**0.9594**	**0.2811**			

Table 2: The confusion matrix for "dataset B" scan. All the scan points are included, not only the medoids of the segments

As you can see in Figure 3(f), after non-associative Markov network classification there remain some errors. But the kinds of errors are different for those methods. While the constant AMN misclassifies whole objects, our method misclassifies parts of them, so such errors could be corrected using some filtering technique. They also might be eliminated by more accurate pairwise potentials tuning.

We also estimate the eligibility of our method for building detection. We consider a building correctly detected if at least 70% of its points are marked as building and form a continuous set. On the "dataset A" all 19 buildings are detected correctly. There are 31 wrong detections, but they could be easily filtered out during post-processing, because they typically consist of a single segment. On the "dataset B" 55 buildings out of 58 are detected correctly along with 63 wrong detections.

4.3 Efficiency

Since we subsample a cloud by means of segmentation, our method runs relatively fast. The timing characteristics are collected in the Table 3. The tests were run on the machine with 2.80 GHz CPU and 1Gb of RAM. Please note that we perform all operations on scans, whose sizes exceed a million points instead of splitting them explicitly, in opposite to, for example, (Munoz et al., 2009b). We build R-Tree, using its leaves as segmentation. There are 26K and 27K segments on "dataset A" and "dataset B", respectively. Then we build the new R-Tree over medoids of the segments, which is the base of a scalable graph construction algorithm.

	dataset A	dataset B
Building index	278	315
Feature computation	100	146
Building graph	351	498
Random Forest classification	8	8
MRF inference	86	60
Total	823	1027

Table 3: Time of computation, sec

5 CONCLUSIONS AND FUTURE WORK

In this paper we proposed the new technique of laser scanning data classification based on non-associative Markov networks. We showed how to perform a tractable MAP inference in a network with non-attractive pairwise potentials and obtained better results than associative Markov network did. Naïve Bayes classifier was used for pairwise potentials tuning just to demonstrate that the method is viable. We believe it is possible to get significant performance improvement by introducing a tailored learning technique. Also, considering higher-order cliques (similar to (Munoz et al., 2009a)) might be fruitful.

ACKNOWLEDGMENTS

The work was partially supported by the federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" and RFBR grant #08-01-00883a.

We would thank Vladislav Kinshakov, the director of NPO "Region" for valuable advices and proofreading, and Dmitry Vetrov for his insightful ideas on usage of MRFs for the task. We also thank Nathalie Chufarova for the help with preparing ground truth data.

REFERENCES

Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G. and Ng, A., 2005. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, pp. 169–176.

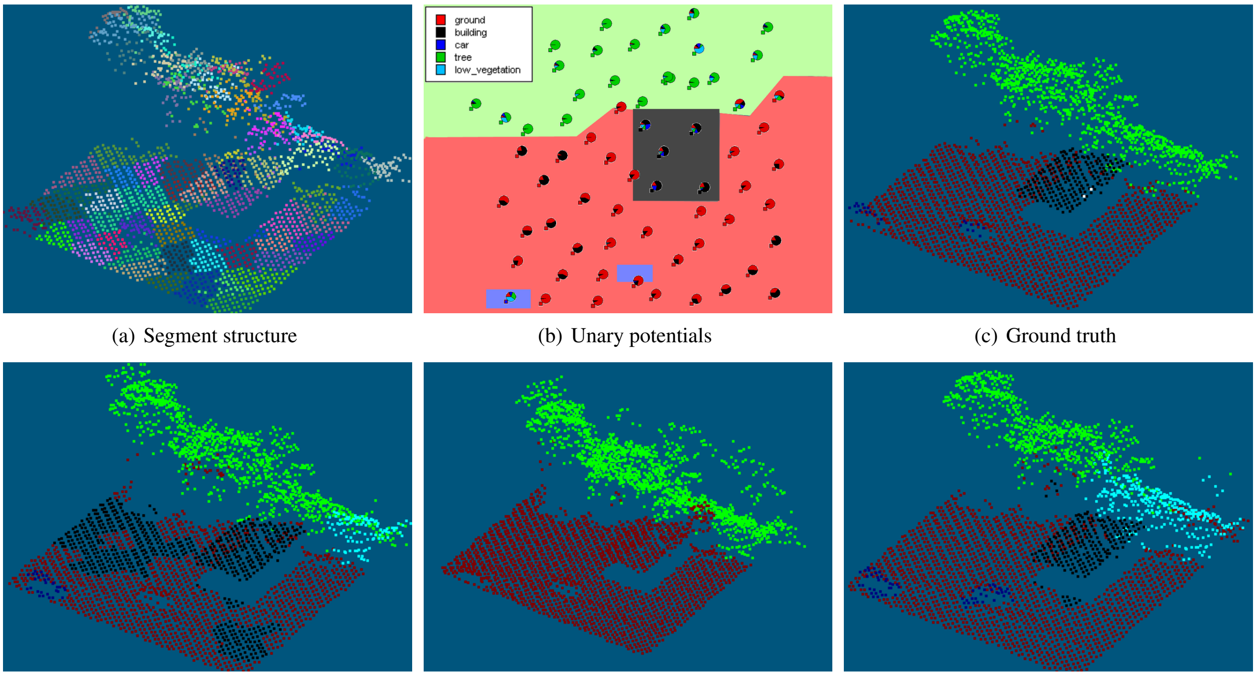
Boykov, Y., Veksler, O. and Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), pp. 1222–1239.

Breiman, L., 2001. Random forests. Machine Learning 45(1), pp. 5–32.

Carlberg, M., Gao, P., Chen, G. and Zakhor, A., 2009. Classifying urban landscape in aerial LiDAR using 3D shape analysis. In: IEEE International Conference on Image Processing, Cairo, Egypt, pp. 1701–1704.

Chehata, N., David, N. and Bretar, F., 2008. LIDAR Data Classification using Hierarchical K-means clustering. The International Archives of Photogrammetry, Remote sensing and Spatial Information Sciences 37, pp. 325–330.

Chehata, N., Guo, L. and Mallet, C., 2009. Airborne lidar feature selection for urban classification using Random Forests. The International Archives of Photogrammetry, Remote sensing and Spatial Information Sciences 38(3/W8), pp. 207–212.



(a) Segment structure

(b) Unary potentials

(c) Ground truth

(d) Random Forest without MRF

(e) Constant AMN

(f) Our method

Figure 3: (a) The output of the segmentation algorithm a piece of “dataset A”. Segments are colored with random colors. (b) The output of Random Forest treated as unary potentials for MRF. Background colour reflect ground truth class label, round histograms reflect the output of multiclass classifier. (c) - (f) Results of the different algorithms. The colours correspond to the classes: red – ground, black – building, deep blue – car, green – tree, cyan – low vegetation. *Better viewed in colour and magnified!*

Elkan, C., 2001. The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, Seattle, WA, pp. 973–978.

Endres, F., Plagemann, C., Stachniss, C. and Burgard, W., 2009. Unsupervised Discovery of Object Classes from Range Data using Latent Dirichlet Allocation. In: Robotics: Science and Systems, Seattle, WA.

Guttman, A., 1984. R-trees: A dynamic index structure for spatial searching. In: ACM SIGMOD International Conference on Management of Data, ACM New York, NY, USA, pp. 47–57.

Joachims, T., Finley, T. and Yu, C., 2009. Cutting-plane training of structural SVMs. Machine Learning 77(1), pp. 27–59.

Johnson, A. and Hebert, M., 1999. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(5), pp. 433–449.

Kohli, P., Kumar, M. and Torr, P., 2007. P3 and Beyond: Solving Energies with Higher Order Cliques. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN.

Kolmogorov, V., 2006. Convergent tree-reweighted message passing for energy minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), pp. 1568–1583.

Kolmogorov, V. and Zabih, R., 2004. What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), pp. 147–159.

Lodha, S. K., Fitzpatrick, D. M. and Helmbold, D. P., 2007. Aerial Lidar Data Classification using AdaBoost. In: International Conference on 3-D Digital Imaging and Modeling, Montreal, pp. 435–442.

Lodha, S. K., Kreps, E. J., Helmbold, D. P. and Fitzpatrick, D., 2006. Aerial LiDAR Data Classification Using Support Vector Machines (SVM). In: International Symposium on 3D Data Processing, Visualization and Transmission, IEEE, Chapel Hill, NC, pp. 567–574.

Lu, W.-L., Murphy, K. P., Little, J. J., Sheffer, A. and Fu, H., 2009. A Hybrid Conditional Random Field for Estimating the Underlying Ground Surface From Airborne LiDAR Data. IEEE Transactions on Geoscience and Remote Sensing 47(8/2), pp. 2913–2922.

Munoz, D., Bagnell, J., Vandapel, N. and Hebert, M., 2009a. Contextual classification with functional Max-Margin Markov Networks. In: IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, pp. 975–982.

Munoz, D., Vandapel, N. and Hebert, M., 2008. Directional associative markov network for 3-d point cloud classification. In: International Symposium on 3D Data Processing, Visualization and Transmission, Atlanta, GA.

Munoz, D., Vandapel, N. and Hebert, M., 2009b. Onboard contextual classification of 3-D point clouds with learned high-order Markov Random Fields. In: IEEE International Conference on Robotics and Automation, Kobe, Japan, pp. 4273–4280.

Sithole, G. and Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 59(1-2), pp. 85–101.

Taskar, B., Chatalbashev, V. and Koller, D., 2004. Learning associative Markov networks. In: International Conference on Machine Learning, Banff, Alberta, Canada, pp. 102–109.

Tovari, D., 2006. Segmentation Based Classification of Airborne Laser Scanner Data. In: PhD thesis, Universitat Fridericiana zu Karlsruhe.

Triebel, R., Kersting, K. and Burgard, W., 2006. Robust 3d scan point classification using associative markov networks. In: IEEE International Conference on Robotics and Automation, Orlando, FL, pp. 2603–2608.

Triebel, R., Shmidt, R., Mozos, O. and Burgard, W., 2007. Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data. In: International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 2225–2230.

Yedidia, J., Freeman, W. and Weiss, Y., 2000. Generalized belief propagation. In: NIPS, Vancouver, Canada, pp. 689–695.