SUPERVISED AND UNSUPERVISED MRF BASED 3D SCENE CLASSIFICATION IN MULTIPLE VIEW AIRBORNE OBLIQUE IMAGES

M. Gerke[a] and J. Xiao[b]

[a]University of Twente, Faculty of Geo-Information Science and Earth Observation – ITC, Department of Earth Observation Science, Hengelosestraat 99, P.O. Box 217, 7500AE Enschede, The Netherlands – m.gerke@utwente.nl
[b]Computer School of Wuhan University, Luoyu Lu 129, Wuhan, P.R. China, 430072 – xiaojingemily@163.com

KEY WORDS: Classification, Graph Cut, Learning, Performance, Point Cloud, Optimization, Random Trees

ABSTRACT:

In this paper we develop and compare two methods for scene classification in 3D object space, that is, not single image pixels get classified, but voxels which carry geometric, textural and color information collected from the airborne oblique images and derived products like point clouds from dense image matching. One method is supervised, i.e. relies on training data provided by an operator. We use Random Trees for the actual training and prediction tasks. The second method is unsupervised, thus does not ask for any user interaction. We formulate this classification task as a Markov-Random-Field problem and employ graph cuts for the actual optimization procedure.

Two test areas are used to test and evaluate both techniques. In the Haiti dataset we are confronted with largely destroyed built-up areas since the images were taken after the earthquake in January 2010, while in the second case we use images taken over Enschede, a typical Central European city. For the Haiti case it is difficult to provide clear class definitions, and this is also reflected in the overall classification accuracy; it is 73% for the supervised and only 59% for the unsupervised method. If classes are defined more unambiguously like in the Enschede area, results are much better (85% vs. 78%). In conclusion the results are acceptable, also taking into account that the point cloud used for geometric features is not of good quality and no infrared channel is available to support vegetation classification.

1 INTRODUCTION AND RELATED WORK

Oblique airborne imaging is entering more and more into photogrammetric production workflows. For a relatively long time Pictometry and the Midas Track'Air system have been a standard in small format multi-head photography, but recently many camera vendors released multi-head midformat camera systems. Examples are *IGI Pentacam, Hexagon/Leica RCD30 Oblique* or *Microsoft Osprey*. While initially the use was for manual interpretation, the stable camera geometry and accurate image orientation procedures enable to perform automated scene analysis. One of the outstanding properties of oblique airborne images is that vertical structures, such as building façades or trees, get depicted. While this is also possible in the border areas of vertical looking nadir images, we have a viewing angle of at least 45° already in the image centre of oblique images. The major shortcoming as a consequence of this property is a large amount of occlusion which needs to be addressed in any automated interpretation method.

In (Gerke, 2011) we demonstrated that for urban scene classification of multiple view airborne images the fusion of radiometric, textural and point cloud-based features in three-dimensional object space showed better results compared to a purely image-based 2D classification. The motivation behind that analysis has been that for oblique images vertical structures within the scene are visible and – opposed to nadir-looking images – can be detected, but because of that the integration in the entire scene must be in 3D space rather than in 2.5D space. We used oblique airborne images over Haiti after it has been severely affected by the earthquake early 2010. A 3D point cloud was derived from the multiple image matching method by Furukawa and Ponce (2010). Features reflecting geometrical, textural and color properties were computed and assigned to voxels representing the scene. While in (Gerke, 2011) we argued that so far 3D scene interpretation was done only rarely and only relied on geometric features, two very interesting and related works appeared in the meantime (Ladický et al., 2012; Haene et al., 2013). Both approaches combine the problem of 3D scene reconstruction and labeling in a joint optimization framework and show some convincing results.

Compared to the aforementioned methods we rely on existing point cloud information, computed from the images beforehand, in this case use state-of-the-art dense matching. This procedure on the one hand splits up the whole problem into two steps, on the other side we can use derived 3D features such as plane normals, their residulas explicitly or normalized heights for the labeling task; in both other works only the pure depth information or image-based features can be exploited for the classification.

The presented former approach (Gerke, 2011) followed a supervised strategy, that is, a human operator needs to provide training data for the voxel-based classification. For many applications, however, an unsupervised and thus fully automatic method is of greater value, e.g. for rapid scene interpretation. The main objective of this paper is to present an extension of the previously defined work towards its embedding in a Markov-Random-Field (MRF) framework, while optimization is carried out through graph cut (Boykov et al., 2001). The approach uses ideas from the optimization-based scene classification introduced by Lafarge and Mallet (2012) but extends this towards the use of color features and the detection of building façades and discrimination of sealed and non-sealed ground objects.

2 DATA PREPARATION

After image processing, i.e. bundle block adjustment and dense image matching (Furukawa and Ponce, 2010), we assume to have proper image orientation and calibration information and a dense point cloud. In a preprocessing stage we perform two steps: 1) filter the ground points and compute normalised heights for non-ground points and 2) convert the point cloud into a voxel representation.

2.1 Ground filtering and height normalization

In order to be able to discriminate ground and off-ground objects we initially label each point whether it is a ground point or not. For this purpose we use the tool *lasground*, which is part of *lastools* (Rapidlasso, 2013). The software largely implements the method proposed by Axelsson (1999), i.e. it is based on mesh simplification. In a subsequent step for each off-ground point the height difference between that point and the closest ground point is computed and stored as the normalized height.

2.2 Spatial enumeration (voxelisation)

The motivation to do a voxelisation of the point cloud is to finally have a more regular point pattern. However, the plane-based point segmentation and features computed from the point cloud (normalized height, normal vectors, see below), are computed from the original data. In this sense the voxels are only carrying the information derived from the points inside a particular cube. The voxel cube side length is defined in order to ensure a good sampling of the original data, so as a rule of thumb the mean ground sampling distance is chosen.

3 METHOD

The workflow of processes within the method we propose is sketched in Fig. 1. The input is given by the point cloud data as derived from image matching, its voxel representation and the original images. Features are computed from the point cloud and from the images and assigned to each voxel. Ultimately we are aiming at classifying segments which are defined by geometric features. To this end we first perform a region growing algorithm, introduced by Vosselman et al. (2004), yielding a segmentation according to planarity of segments. All voxels not assigned to a planar segment are clustered through a connected components analysis and each cluster is treated as independent segment in the following. The segmentation information is then assigned to the voxels and feature values are computed per segment. We use two different independent classification schemes: a supervised approach, based on Random Trees (RTree, (Breiman, 2001)), and an unsupervised, rule-based approach which applies a graph cut-based optimization scheme.

3.1 3D geometry- and image-based features

Points from image matching explicitly represent the geometry of objects. Man-made objects are mainly composed out of planar faces, as opposed to natural surfaces like trees and shrub. In addition the height above the ground surface helps to distinguish building roofs and trees from ground. So, beside the normalized height, computed in the pre-processing step, we estimate per point the normal of a face, composed out of the 10 closests points. In particular we use the residual of the normal, which helps to distinguish smooth from rough surfaces. The residual of the normal corresponds to the smallest eigenvalue of the covariance matrix associated with the centre of gravity, computed from all points under consideration; the normal vector is the corresponding eigenvector. The Z-component of the normal is used as well in order to distinguish horizontal from vertical and other planes.

Concerning image-based features we compute color values (Hue, Saturation, in this case), texture in the form of a standard deviation in a 9x9 matrix around each image pixel, and straight lines. The latter one is computed using the line growing algorithm by Burns et al. (1986). For each image pixel which is part of a straight line we encode the length of that line as feature, associated to that pixel. We use straight lines because our assumption is that at man-made structures, such as roofs or road surfaces we find linear structures and thus the incorporation of such an information into the classification will help to distinguish man-made from natural objects.

We use a visibility analysis which checks for every voxel if the line of sight between that voxel and the projection centre of the respective camera is blocked by any other voxel. In order to avoid effects of void areas caused by an insufficient matching performance the voxels used for this test are dilated by factor 10. Details of this method can be found in (Nyaruhuma et al., 2012). Thanks to this visibility check we ensure that image based features are assigned to the correct voxel when we back project the voxel to image space. Since we have overlapping images, values for a certain feature will be observed in multiple images. Therefore, the final feature value will be computed from the median of all input values.

3.2 Combination of features per segment

We use the segment as the main entity for classification, so we need to compute per feature a joint value representing all voxels inside this segment. To this end we compute a mean value per feature, associated to each segment. Further we compute a standard deviation which is used as weight in the optimization-based classification. To summarize the following features are available per segment: The *normalized height* helps to distinguish ground from non-ground segments, the *z-component of plane normal* helps to distinguish horizontal from slanted or vertical planes (façades), the *residual of plane normal* is a measure for segment roughness. In addition we use *2 color features*: hue, saturation, the *standard deviation in a 9x9 window*, which is a texture measure and related to surface properties, and finally we use *straight line length and direction* which provide evidence for man-made structures.

3.3 Supervised segment classification using Random Trees

One of the objective of this paper is to compare the performance of a supervised classification approach and an optimization technique which does not need training information, using the same features.

Two state-of-the-art machine learning techniques which showed good results in our previous experiments (Gerke, 2011) are adaptive boosting ”AdaBoost” (Freund and Schapire, 1996) and Random Trees (Breiman, 2001). Because of the similar results obtained earlier we believe that the actual selection of a supervised method is not very critical here, so we chose the RTrees approach for the processing to leave space for the actual comparison between supervised and un-supervised methods in the result section.

In a manual processing step an operator creates reference data by labeling the original images. The labels are then transferred to the point cloud through a simple back projection of the 3D points to the images, only considering the actually visible points in the respective image. The feature vector per segment and the reference class are then fed into a RTree learning scheme. In order to monitor the quality of learning (e.g. to detect overfitting) the training and prediction is done several times, where each time a different subset (about 20%) of reference data is used for the training. Since in later experiments no significant difference showed up between the single runs, only the first result is used in later sections to simplify the analysis.

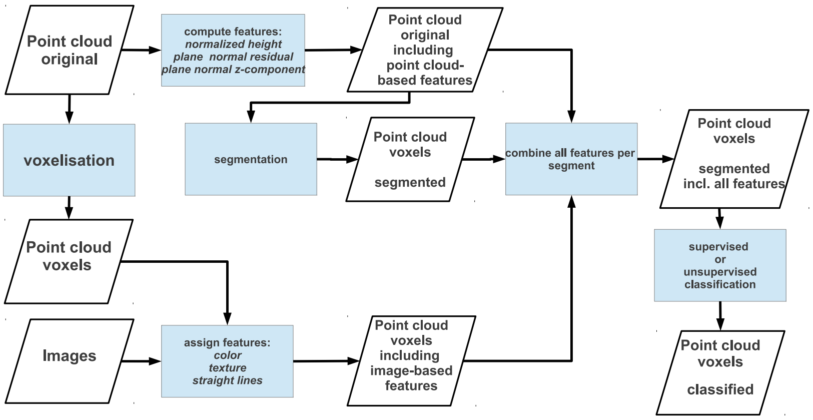
Point cloud original
compute features: normalized height plane normal residual plane normal z-component
Point cloud original including point cloud-based features
voxelisation
segmentation
Point cloud voxels segmented
combine all features per segment
Point cloud voxels segmented incl. all features
Point cloud voxels
supervised or unsupervised classification
Images
assign features: color texture straight lines
Point cloud voxels including image-based features
Point cloud voxels classified

Figure 1: Entire workflow from point clouds and images to classification

3.4 Fully automatic classification in a MRF framework

The advantage of this scheme over other techniques is that it combines observations (data term) with a neighborhood smoothness constraint. Among a selection of optimization methods to minimize the overall energy, the graph cut (Boykov et al., 2001) approach showed good performance in the past. The representation is in the 3D lattice, i.e. each voxel will retrieve an individual label, and thus also an individual data term and the neighborhood is defined in the 3D lattice, as well. However, in order to represent the segmentation the features as computed per segment will be used for the respective voxels inside. We compute energy data terms for the classes *roof, tree, façade, vegetated ground, sealed ground and roof destroyed/rubble* and also add a class *background* to represent ”empty” voxels. In the following the class *roof destroyed/rubble* will be named *rubble*.

The total E energy is composed out of the data term and a pairwise interaction term:

$$E \quad = \quad \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V_{pq}(f_p, f_q), \qquad (1)$$

where $D_p(f_p)$ is the data energy at point (i.e. voxel) p for class f_p. V_{pq} is the pairwise interaction potential, considering the neighborhood N. In particular we use the Potts interaction potential as proposed by Boykov et al. (2001) which adds a simple label smoothness term:

$$V_{pq}(f_p, f_q) = \lambda_{pq} T, \text{ with}$$
$$T = \begin{cases} 0, & \text{if } f_p = f_q \text{ and} \\ 1, & \text{else} \end{cases}$$

Features contributing to the data term are normalized to the range $[0, 1]$. Internally all features are stored in 8bit images, i.e. in a resolution of $1/256$. Especially for height values this is the reducing factor, and needs to be taken into account. The feature values contribute to factors for the total energy, depending on the actual class. Each factor $S_\square$ is initialised with 1 to avoid that in case a feature does not contribute evidence for a particular class the total energy vanishes. The features and the derived factors are defined as follows:

Normalized Height m_H: the normalized height coded in decimeter dm. In order to reflect the fact that ground objects have a low height, and – depending on the object type – roofs or trees have a significantly large height above terrain we compute some values $m_{Hx} = |m_H - x|$. The fact that we can store only 256 different values means that we can represent normalized heights up to $25.5m$. All heights above this value are set to that maximum. For our task this does not restrict the functionality since the normalized height is basically used to differentiate ground from non-ground features only.

$$S_H = \begin{cases} \min(m_{H30}, m_{H60}, m_{H90}), & \text{if } f_p\text{=roof} \\ m_{H30}, & \text{if } f_p \in \{\text{tree,façade}\} \\ \min(m_{H5}, m_{H30}), & \text{if } f_p\text{=rubble} \\ m_{H5}, & \text{else} \end{cases}$$

Note: The heigt is defined in dm, hence the energy formulation for roofs is in this example best for buildings up to $9m$, but can easily extended for taller buildings. For rubble we cannot really use knowledge concerning its normalized height. It can be close to ground, or form a relatively tall rubble pile.

Line Length m_{L-} and m_{L+}: if one or more lines are assigned to the segment where the voxel is located, we compute two different values: m_{L-} the difference to the shortest line in the overall area and m_{L+}: the difference to the longest line. Those two values are used in the energy computation, depending on the class assuming that longer lines can be found at man-made structures, shorter lines in natural environments or at destroyed buildings.

$$S_L = \begin{cases} m_{L+}, & \text{if } f_p \in \{\text{roof,façade}\} \\ m_{L-}, & \text{else} \end{cases}$$

Plane Normal, Z-component m_Z: to distinguish horizontal from vertical and other planes

$$S_Z = \begin{cases} m_Z, & \text{if } f_p\text{=façade} \\ 1 - m_Z, & \text{if } f_p\text{=sealed\_grnd} \\ \min(1 - m_Z, |0.5 - m_Z|), & \text{if } f_p\text{=roof} \\ \min(m_Z, 1 - m_Z, |0.5 - m_Z|) + C, & \text{else} \end{cases}$$

Note: For *tree, vegetated ground* and *rubble* the normal vector cannot contribute any evidence – it is arbitrary. In order to avoid that this ignorance has an influence on the total energy, the factor for those classes is the same as the minimum energy from the other classes, with an added very small constant energy C.

Texture: Standard Deviation m_T: standard deviation in a sliding 9x9 window in an image. Since we assume a large value for trees we also compute the overall maximum standard deviation m_{Tmax}. For tree and rubble we expect larte m_T values.

$$S_T = \begin{cases} |m_T - m_{Tmax}|, & \text{if } f_p \in \{\text{tree,rubble}\} \\ m_T, & \text{else} \end{cases}$$

Color m_C: We use the RGB values to compute the saturation (SAT) and HUE values. We observed that the saturation is generally high for vegetation, while it is low for sealed areas and rubble:

$$S_{C_S} = \begin{cases} 1 - m_{C_S}, & \text{if } f_p \in \{\text{tree,veg\_grnd}\} \\ m_{C_S}, & \text{else} \end{cases}$$

In the HUE definition the color green is defined at 120°. Hence we assume that vegetation has a peak around that hue value, while other classes show a relatively small signal there:

$$S_{C_H} = \begin{cases} 1 - |m_{C_H} - 120^*|, & \text{if } f_p \in \{\text{tree,veg\_grnd}\} \\ \min(1 - |m_{C_H}|, 1 - |m_{C_H} - 240|), & \text{else} \end{cases}$$

*: For the sake of simplicity the fixed angular values are given in the original unit. In practice they are also scaled to [0,1]. -

To consider the uncertainty inherent in the feature values resulting from the merging of values from different views we add a to *each factor* a penalizing energy which is proportional to the standard deviation computed during the merge of feature values per segment.

In conclusion each feature contributes to a final factor $S'_\square = S_\square + S_{\square_{pen}}$ per object class, which is defined in [0,1]. The energy computed from all features voting for a certain label p is

$$\begin{aligned} D_p(f_p) = \quad & S'_H(f_p) \cdot S'_L(f_p) \cdot S'_Z(f_p) \cdot S'_T(f_p) \cdot \\ & S'_{C_S}(f_p) \cdot S'_{C_H}(f_p) \end{aligned}$$

In the practical implementation of graph cut we need to convert the energies into integer values. Some internal experiments showed that an actual ranking of the energy per entity gives the best result. That is, we assign an energy defined in $[0, N-1]$ – where N is the number of classes – to $D_p(f_p)$, according to the sequence in the original $D_p(f_p)$ computation.

4 RESULTS

4.1 Haiti test area

In our experiments we concentrated on the same test area as described in (Gerke and Kerle, 2011; Gerke, 2011). Pictometry[1] images were acquired over Port-au-Prince, Haiti, in January 2010 a few days after the earthquake. The ground sampling distance (GSD) varies from $10cm$ to $16cm$ (fore- to background). Due to varying image overlap configuration the number of images which observe a particular part of the scene varies from 4 to 8. In this experiment only oblique images were used since the nadir images are shipped only after ortho rectification, and without further specification of the ortho image production process. See Fig. 2 for some example images and results. Opposed to the experiments conducted earlier we are now using more images (11 instead of only 3). Earlier we only used three images to directly compare per-image to object space classification. Here we concentrate only on the object space and thus exploit all the information we have.

4.1.1 RTrees result The confusion table showing the RTrees classification per-segment result using the validation subset of the reference data is given in Table 1. Percentages refer to the total number of reference entities, i.e. rows sum up to 100% (±because of round-off errors). The overall classification accuracy – computed as the normalized trace of the confusion matrix – is 73.1%. For the least clearly definable classes *rubble* and *sealed_grnd* we observe a strong interclass confusion: actually almost 29% of the *sealed_grnd* got classified as *rubble*. This has certainly got to do with the varying height levels of *rubble*. This statement is also supported by the observation that approximately 14% of *rubble* segments got labeled *roof*. Other interclass confusion worth mentioning here concerns façades and roofs: 10% of all *façade* segments got labeled as roof. Typically this misclassification occurs at slanted roofs.

REF→	Facade	Roof	Rubble	Seal_Grd	Tree
Facade	0.721	0.105	0.093	0.070	0.012
Roof	0.046	0.796	0.086	0.060	0.011
Rubble	0.078	0.143	0.681	0.076	0.023
Seal_Grd	0.000	0.048	0.286	0.619	0.048
Tree	0.049	0.049	0.024	0.024	0.854

Table 1: Confusion matrix RTrees Haiti, overall accuracy 73.1%

Compared to the result published earlier we obtain similar results for all classes except for *tree*, which has a correctness here of 85% while it was much worse earlier (around 25% only). Two reasons might explain this. First, in the earlier work we did not use the normalized height explicitly, and second, the use of multiple images renders more trees visible as before.

4.2 Unsupervised result

The per-voxel result from the unsupervised, MRF-based method for the Haiti testdata is shown in the confusion matrix in Table 2. Here the problem of the unclear object class definition for *rubble* becomes even more obvious than in the supervised result. Each object class got labeled as *rubble* by at least 24% (building), and up to 67% (sealed ground). Another reason for classification errors is relating to the height normalization, and in particular the ground filtering. Especially in destroyed areas, for instance at rubble piles, there is a kind of smooth transition between ground and non-ground features, and the latter ones might then get labeled as ground. This is observable in the results as well: actually 12% of all roof voxels got labeled as *sealed_grnd*. The major differentiation in the energy function for those two classes is made from the normalized height. In the result from the supervised method that problem does not become this obvious since the height uncertainty gets implicitly modeled during RTrees training.

REF→	Facade	Roof	Rubble	Seal_Grd	Tree
Facade	0.665	0.042	0.243	0.030	0.021
Roof	0.097	0.441	0.344	0.118	0.001
Rubble	0.031	0.084	0.836	0.044	0.006
Seal_Grd	0.023	0.004	0.671	0.267	0.035
Tree	0.008	0.042	0.356	0.009	0.586

Table 2: Confusion matrix unsupervised through MRF/graph cut Haiti, overall accuracy 58.9%

4.3 Enschede test area

To use the Haiti data for the experiments has the drawback that object classes in the scene after a major seismic event are not as clearly definable as in an intact region. For this reason we will also demonstrate the new method in a dataset showing a typical European sub-urban scene. The images over Enschede were acquired by Slagboom en Peeters[2] in May 2011. This company mounted five Canon EOS 5D Mark II cameras in one head, using the same *Maltese Cross* configuration as Pictometry: one camera pointing into nadir direction and the remaining ones into the four cardinal directions under a tilt angle of 45°. Due to the low flight altitude of $500m$ above ground the GSD varies from 5 to 8 cm, and the image overlap is at least 60% in all directions, i.e. making

[1] http://www.pictometry.com

[2] http://www.slagboomenpeeters.com/

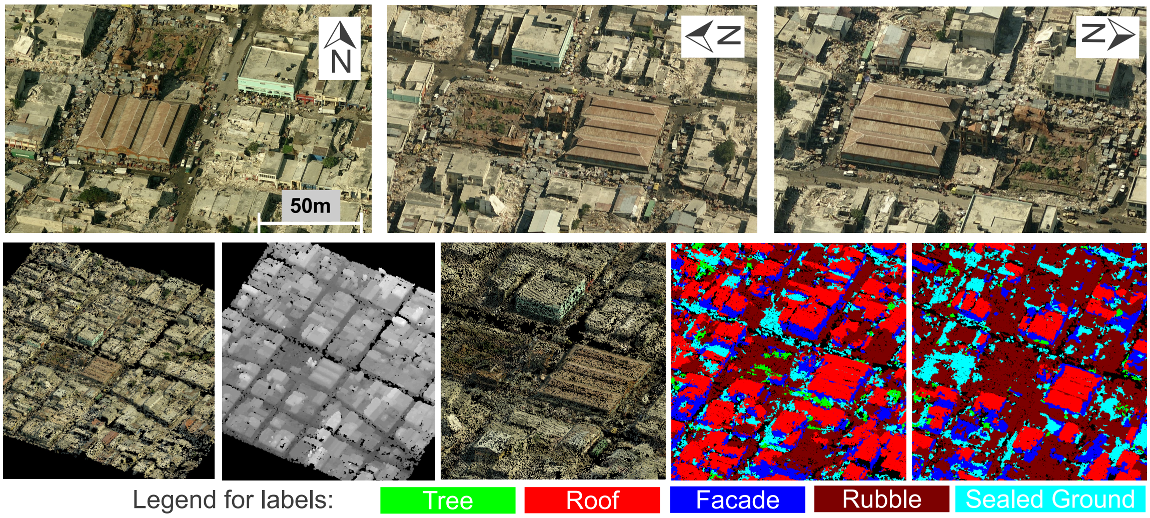

Figure 2: Example oblique images from Port-au-Prince and derived information. Upper row: oblique images facing North, East, West direction. Lower row: colored point cloud computed with PMVS2 (Furukawa and Ponce, 2010), the same with grey values representing height, zoom in to an example area, result from supervised classification, result from MRF-based classification. Images: ©Pictometry, Inc.

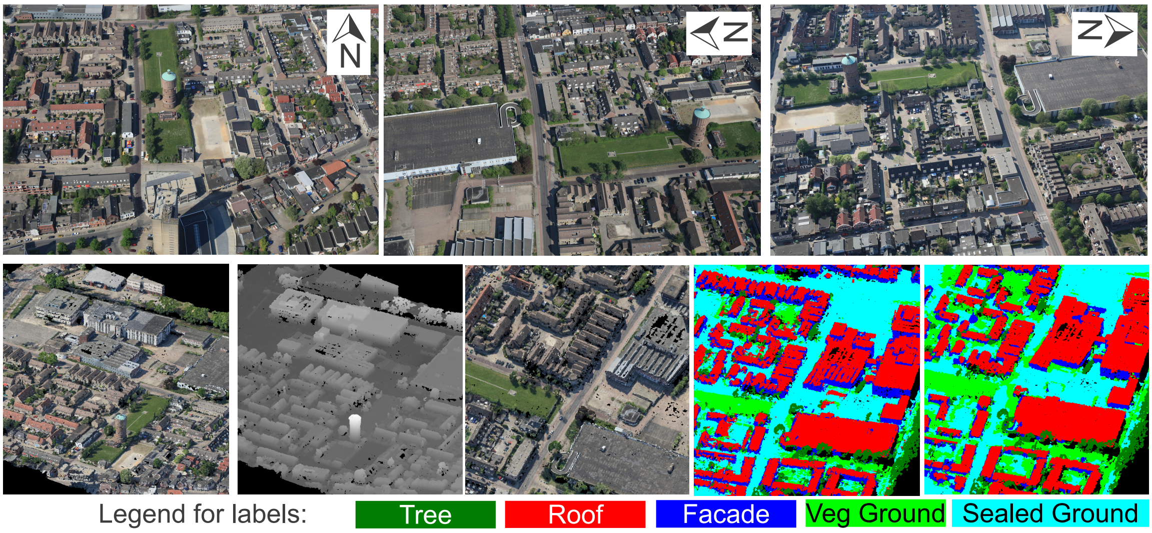

Figure 3: Example oblique images and derived information for Enschede. Upper row: oblique images facing North, East, West direction. Lower row: colored point cloud computed with PMVS2, the same with grey values representing height, zoom in to an example area, result from supervised classification, result from MRF-based classification. Images: ©Slagboom en Peeters B.V.

stereo overlap available from all perspectives. Residuals at check points are quite large, they are in the range of more than 2 pixels (RMSE up to $20cm$), which also has an effect on the point cloud computation; the standard deviation computed from plane fitting residuals partly goes up to $50cm$, see (Xiao, 2013) for a more detailed evaluation. In Fig. 3 some example images showing the area are displayed in the upper row. The scene shows some quite large industrial halls, but also a typical settlement area with individual free standing houses of different construction type. The bottom row shows the colored point cloud of the entire test area, first real colors, then a gray coded height visualisation. The centre image shows a zoom-in to a region of interest and the two right hand images give again the classification result (RTrees next to the middle image, MRF-based right hand).

4.3.1 RTrees result The confusion matrix in Table 3 shows better overall accuracy compared to the Haiti dataset; it is almost 85%. Remarkable is again the confusion of roofs and façades: 12% of façades got labeled as roof which is mainly due to small structures at facades like extensions with a flat roof structure, which are *façade* in the reference, but got classified as *roof*. Moreover segments in the upper part of façades are often classified as *roof* due to the height. Another main confusion concerns trees and ground vegetation: more than 14% of tree-segments got labeled *veg_grnd*. In this case the quite fuzzy definition of those classes causes most of the confusion: close-to-ground vegetation like bushes is labeled by the operator as ground vegetation, however, due to their elevation the classifier might confuse it with trees. Since we do not have access to infrared channels in this case the classifier cannot make use of spectral information to further separate those classes. This is also the reason for a certain interclass-confusion between sealed and vegetated ground segments.

4.4 Unsupervised result

The overall classification accuracy for our unsupervised, MRF-based approach is 78.3%, i.e. some 6% worse compared to the supervised method, but by 20% better as for the Haiti data. Looking closer at the confusion matrix in Table 4 reveals similar ten-

REF→	Facade	Roof	Seal_Grd	Veg_Grd	Tree
Facade	0.806	0.122	0.054	0.002	0.015
Roof	0.075	0.876	0.036	0.001	0.013
Seal_Grd	0.032	0.009	0.905	0.043	0.012
Veg_Grd	0.000	0.000	0.069	0.862	0.069
Tree	0.019	0.019	0.049	0.143	0.771

Table 3: Confusion matrix RTrees Enschede, overall accuracy 84.7%

dencies as for supervised case, but the interclass confusions are in general larger. The mix-up of roofs and façades on the one hand and trees and ground vegetation on the other hand is significant here as well. Since the height variations are not modeled in the MRF energy definition, the confusion is even larger than in the supervised case. The same holds for the modeling of color features to support the discrimination between sealed and vegetated ground.

REF→	Facade	Roof	Seal_Grd	Veg_Grd	Tree
Facade	0.628	0.231	0.073	0.051	0.018
Roof	0.066	0.838	0.071	0.006	0.020
Seal_Grd	0.002	0.001	0.800	0.196	0.000
Veg_Grd	0.000	0.000	0.029	0.968	0.003
Tree	0.012	0.045	0.037	0.303	0.602

Table 4: Confusion matrix unsupervised through MRF/graph cut Enschede, overall accuracy 78.3%

5 DISCUSSION AND CONCLUSIONS

We have developed and tested two different 3D scene classification methods; one was a supervised scheme, based on the Random Trees machine learning technique. The second one was formulated in a MRF framework, where the graph cut approach was used for energy minimization. All in all most of the results can be considered satisfactory, however, there are some specific problems. If object classes are not clearly defined, that is they significantly share properties with other classes, like rubble in our example, the MRF-based method basically fails. The relatively good result for the same method, but in a better structured environment shows that such unsupervised optimization method are applicable in real-world scenarios. In fact, in that case the overall accuracy is only worse by 6% compared to the supervised method. We assume that the quite noisy point cloud from the image matching has also an impact on the classification quality, at least for the unsupervised method. Thus in further work we need to have a closer look into that. A problem inherent in the approach is its overall dependency on the 3D point cloud accuracy and completeness: if parts of the scene are not represnted in the pointcloud, e.g. because of poor texture, the respective area does not get considered at all. This is a drawback and would need some attention in the future.

Apart from that the whole classification will potentially become more accurate in the future. Upcoming multiple camera heads like Hexagon/Leica RCD30 Oblique or Microsoft Osprey will have NIR channels available which will make the vegetation classification much simpler as in our case where only RGB was available. In addition the mid-frame cameras used in those systems are supposed to have a more stable camera geometry and better lenses compared to the DSLR cameras employed here.

ACKNOWLEDGEMENTS

Pictometry Inc. provided the dataset over Haiti. Slagboom en Peeters B.V. provided the Enschede data. We made use of the implementation of graph cut optimization by Boykov and Kolmogorov (2004). We thank the anonymous reviewers for their valuable comments.

References

Axelsson, P., 1999. Processing of laser scanner data - algorithms and applications. ISPRS Journal of Photogrammetry and Remote Sensing 54(2-3), pp. 138–147.

Boykov, Y. and Kolmogorov, V., 2004. An experimental comparison of min-cut/max- flow algorithms fokr energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), pp. 1124–1137.

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.

Burns, J., Hanson, A. and Riseman, E., 1986. Extracting straight lines. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(4), pp. 425–455.

Freund, Y. and Schapire, R. E., 1996. Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kauman, p. 148156.

Furukawa, Y. and Ponce, J., 2010. Accurate, dense, and robust multi-view stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(8), pp. 1362–1376.

Gerke, M., 2011. Supervised classification of multiple view images in object space for seismic damage assessment. In: Lecture Notes in Computer Science: Photogrammetric Image Analysis Conference 2011, Vol. 6952, Springer, Heidelberg, pp. 221–232.

Gerke, M. and Kerle, N., 2011. Automatic structural seismic damage assessment with airborne oblique pictometry imagery. Photogrammetric Engineering and Remote Sensing 77(9), pp. 885–898.

Haene, C., Zach, C., Cohen, A., Angst, R. and Pollefeys, M., 2013. Joint 3d scene reconstruction and class segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ladický, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W. and Torr, P., 2012. Joint optimization for object class segmentation and dense stereo reconstruction. International Journal of Computer Vision 100(2), pp. 122–133.

Lafarge, F. and Mallet, C., 2012. Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation. International Journal of Computer Vision 99(1), pp. 69–85.

Nyaruhuma, A. P., Gerke, M., Vosselman, G. and Mtalo, E. G., 2012. Verification of 2D building outlines using oblique airborne images. ISPRS Journal of Photogrammetry and Remote Sensing 71, pp. 62–75.

Rapidlasso, 2013. Homepage lastools. http://lastools.org.

Vosselman, G., Gorte, B., Sithole, G. and Rabani, T., 2004. Recognising structure in laser scanner point clouds. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36number 8/W2, pp. 33–38.

Xiao, J., 2013. Automatic building detection using oblique imagery. PhD thesis, University of Twente Faculty of Geo-Information and Earth Observation (ITC), Enschede.