Automated Super-Voxel Based Features Classification of Urban Environments by Integrating 3D Point Cloud and Image Content

Pouria Babahajiani [#1], Lixin Fan [*2], Joni Kamarainen [#3], Moncef Gabbouj [#4]

[#] *Department of Signal Processing, Tampere University of Technology, Tampere, Finland;*

[1] pouria.babahajiani@tut.fi

[3] joni.kamarainen@tut.fi

[4] Moncef.gabbouj@tut.fi

[*] *Nokia Research Centre, Tampere, Finland;*

[2] fanlixin@ieee.org

Abstract—**In this paper we present a novel street scene semantic recognition framework, which takes advantage of 3D point cloud captured by a high definition LiDAR laser scanner. An important problem in object recognition is the need for sufficient labeled training data to learn robust classifiers. We show how to significantly reduce the need for manually labeled training data by reduction of scene complexity using non-supervised ground and building segmentation. Our system first automatically segments grounds point cloud. Then, using binary range image processing building facades will be detected. Remained point cloud will grouped into voxels which are then transformed to super voxels. Local 3D features extracted from super voxels are classified by trained boosted decision trees and labeled with semantic classes e.g. tree, pedestrian, car. Given labeled 3D points cloud and 2D image with known viewing camera pose, the proposed association module aligned collections of 3D points to the groups of 2D image pixel to parsing 2D cubic images.**

Index Term—**3D point cloud, LiDAR, Classification, Segmentation, Image alignment, Street view, Feature extraction, Machine learning, Mobile laser scanner**

I. INTRODUCTION

Analysis of 3D spaces comes from the demand to understand the environment surrounding us and to build more and more precise virtual representations of that space. While many image and video processing techniques for 2 dimensional object recognition have been proposed [1, 2], the accuracy of these systems is to some extent unsatisfactory because 2D image cues are sensitive to varying imaging conditions such as lighting, shadow and etc. In order to alleviate sensitiveness to different image capturing conditions, many efforts have been made to employ 3D scene features derived from single 2D images and thus achieving more accurate object recognition [3]. For instance, when the input data is a video sequence, 3D cues can be extracted using Structure From Motion (SFM) techniques [4]. In the last recent decade, as the 3D sensors begun to spread and the commercially available computing capacity has grown big enough to be sufficient for large scale 3D data processing, new methods and applications were born. Since such 3D information is invariant to lighting and shadow, as a result, significantly more accurate parsing results are achieved.

While a laser scanning or LiDAR (Laser Illuminated Detection And Ranging) system provides a readily available solution for capturing spatial data in a fast, efficient and highly accurate way, the enormous volume of captured data often come with no semantic meanings. Some of these devices output several million data points per second. Efficient, fast methods are needed to filter the significant data out of these streams or high computing power is needed to post-process all this large amount of data. We, therefore, develop techniques that significantly reduce the need for manual labelling of data and apply the technique to the all datasets. Automatic urban environments objects recognition refers to the process of segmentation and classification of objects of interest into predefined semantic labels such as building, tree or car etc. This task is often done with a fixed number of object categories, each of which requires a training model for classification scene components.

The researches related to 3D urban scene analysis had been often performed using 3D point cloud collected by airborne LiDAR for extracting vegetation and building structures [5]. Hernndez and Marcotegui use range images from 3D point clouds in order to extract k-flat zones on the ground and use them as markers for a constrained watershed [6].Recently, classification of urban street objects using data obtained from mobile terrestrial systems has gained much interest because of the increasing demand of realistic 3D models for different objects common in urban era. A crucial processing step is the conversion of the laser scanner point cloud to a voxel data structure, which dramatically reduces the amount of data to process. Yu Zhou and Yao Yu (2012) present a voxel-based approach for object classification from TLS data [7].

It is a challenging task to develop such a vision system because of limitations of sensor model, noise in the data, clutter, occlusion and self-occlusion. In a realistic street scene, there are number of moving objects: people, vehicles, some vegetation. All the points belonging to these objects change from frame to frame. A false registration can easily occur when the algorithm falsely detects moving points as

978-1-4799-8996-6/15/$31.00 ©2015 IEEE

significant points and tries to align consecutive point clouds along these moving points.

The contribution of this work are as follows:

- A complete scene parsing system is devised and experimentally validated using 3D urban scenes that have been gathered with different type of LiDAR acquisition devices. The steps such as segmentation, feature extraction, voxelization are generic and adaptable to solve object class recognition problems in different streets with varying landscape.
- Proposed two-stage (supervised and non-supervised) classification pipeline which requires only small amount of time for training.
- Propose to use novel geometric features leads to more robust classification results
- Using LiDAR data aligned to image plane leads to segmentation algorithm which is robust to varying imaging condition.
- Develop a novel street object recognition method which is robust to different types of LiDAR point clouds acquisition methods.

II. PROPOSED METHODOLOGY

We take a hybrid two-stage approach to address the above mentioned challenges. Firstly, we adopt an unsupervised segmentation method to detect and remove dominant ground and buildings from other LiDAR data points, where these two dominant classes often correspond to the majority of point clouds. Secondly, after removing these two classes, we use a pre-trained boosted decision tree classifier to label local feature descriptors extracted from remaining vertical objects in the scene. This work shows that the combination of unsupervised segmentation and supervised classifiers provides a good trade-off between efficiency and accuracy. The output of classification phase is 3D labeled point cloud and each point is labeled with a predefined semantic classes such as building, tree, pedestrian and etc.

Given a labeled 3D point cloud and 2D cubic images with known viewing camera pose, the association module aims to establish correspondences between collections of labeled 3D points and groups of 2D image pixels. Every collection of 3D points is assumed to be sampled from a visible planar 3D object i.e. patch and corresponding 2D projections are confined within a homogenous region i.e. SuperPixels (SPs) of the image. The output of the 2D-3D alignment phase is 2D segmented image, in which every pixel is labeled based on 3D cues. In contrast to existing image-based scene parsing approaches, the proposed 3D LiDAR point cloud based approach is robust to varying imaging conditions such as lighting and urban structures.

The framework of the proposed methology is given in figure 1, in which 3D LiDAR point cloud and cubic images are the inputs of the processing pipeline and parsing results are presented as 3D labeled point cloud and 2D image segmented with different class labels e.g. Building, road, car and etc.

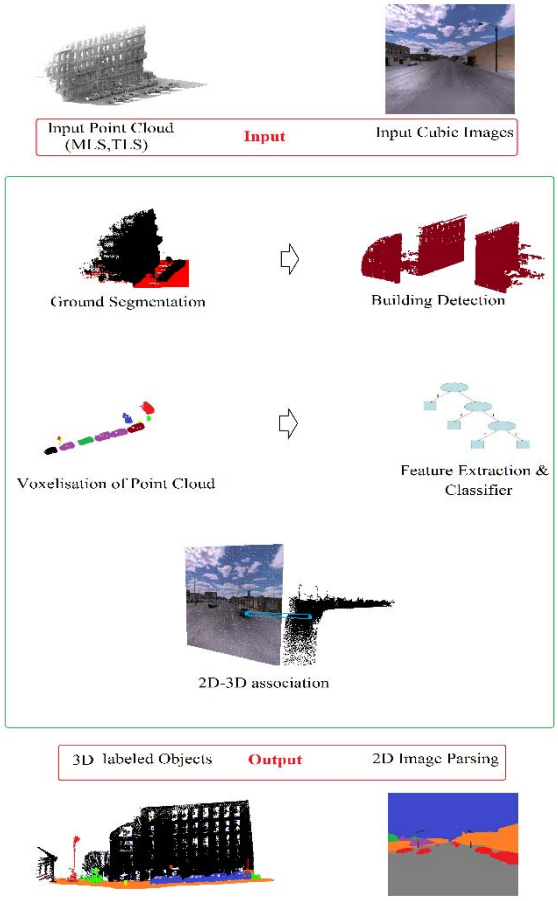

Fig 1. Framework of proposed methology

A. Ground Segmentation

Given a 3D point cloud of an urban street scene, the proposed approach starts by finding ground points by fitting a ground plane to the scene. This is because the ground connects almost all other objects and we will use a connect component based algorithm to over-segment the point clouds in the following step. The plane RANSAC fitting method is used to approximate ground section of the scene. The RANSAC algorithm was developed by Fischler et al. [8] and is used to provide a more robust fitting of a model to input data in the presence of data outliers. Given a 3D point cloud of an urban street scene, the scene point cloud is first divided into sets of 10m×10m regular, non-overlapping tiles along the horizontal x–y plane. Then the following ground plane fitting method is repeatedly applied to each tile. We assume that ground points are of relatively small z values as compared to points belonging to other objects such as buildings or trees (see Fig 2).

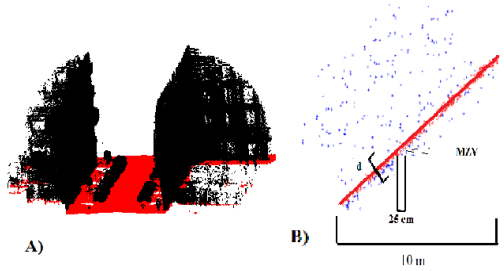

Fig 2. Ground Segmentation. A) Segmented ground and remained vertical objects point cloud are illustrated by red and black colour respectively. B) Sketch map of fitting plane to one tile

The ground is not necessarily horizontal, yet we assume that there is a constant slope of the ground within each tile. Therefore, we first find the minimal-z-value (MZV) points within a multitude of 25cm×25cm grid cells at different locations. For each cell, neighbouring points that are within a z-distance threshold from the MZV point are retained as candidate ground points. Subsequently, a RANSAC method is adopted to fit a plane to candidate ground points that are collected from all cells. Finally, 3D points that are within certain distance (d in Figure 2, B) from the fitted plane are considered as ground points of each tile. The approach is fully automatic and the change of two thresholds parameters do not lead to dramatic change in the results. On the other hand, the setting of grid cell size as 25cm×25cm maintains a good balance between accuracy and computational complexity.

B. Building Segmentation

Our method automatically extract building point cloud (e.g. doors, walls, facades, noisy scanned inner environment of building) based on two assumptions: a) building facades are the highest vertical structures in the street; and b) other non-building objects are located on the ground between two sides of street. As can be seen in figure 3, our method projects 3D point clouds to range images because they are convenient structures to process data. Range images are generated by projecting 3D points to horizontal x–y plane. In this way, several points are projected on the same range image pixel. We count the number of points that falls into each pixel and assign this number as a pixel intensity value. In addition, we select and store the maximal height among all projected points on the same pixel as height value. We define range images by making threshold and binarization of I, where I pixel value is defined as equation 1:

$$I_i = \frac{P_{intensity}}{\text{Max}\_P_{intensity}} + \frac{P_{\text{height}}}{\text{Max}\_P_{\text{height}}} \qquad (1)$$

Where I_i is grayscale range image pixel value, $P_{intensity}$ and P_{height} are intensity and height pixel value and $Max\_P_{intensity}$ and Max_P_{height} represent the maximum intensity and height value over the grayscale image. On the range image, an interpolation is required in order to fill holes caused by occlusions, missing scan lines and LiDAR back projection scatter.

In the next step we use morphological operation (e.g. close and erode) to merge neighbouring point and filling holes in the binary range images (see middle image in Fig 3). The morphological interpolation does not create new regional maxima, furthermore it can fill holes of any size and no parameters are required. Then we extract contours to find boundaries of objects. In order to trace contours, Pavlidis contour-tracing algorithm [9] is proposed to identify each contour as a sequence of edge points. The resulting segments are checked on aspects such as size and diameters (height and width) to distinguish building from other objects. More specifically, equation (2) defines the geodesic elongation E(X), introduced by Lantuejoul and Maisonneuve (1984), of an object X, where S(X) is the area and L(X) is the geodesic diameter.

$$\text{E}(\pi) = \frac{\pi\, L^2\,(X)}{4S(X)} \qquad (2)$$

Considering the sizes and shape of buildings, the extracted boundary will be eliminated if its size is less than a threshold. The resolution of range image is the only projection parameter during this point cloud alignment that should be chosen carefully. If each pixel in the range image cover large area in 3D space too many points would be projected as one pixel and fine details would not be preserved. On the other hand, selecting large pixel size compared to real world resolution leads to connectivity problems which would no longer justify the use of range images. In our experiment, a pixel corresponds to a square of size .05 m^2.

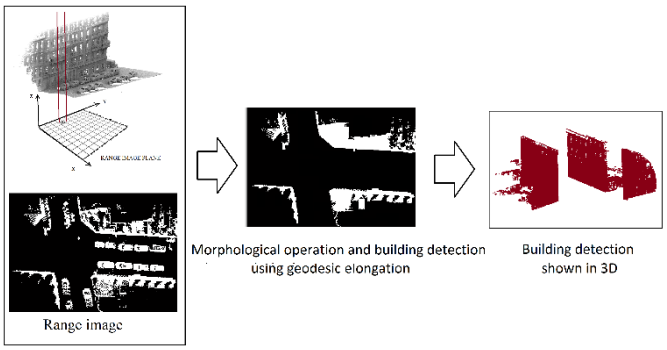

Fig 3. Building Segmentation

The 2D image scene is converted back to 3D by extruding it orthogonally to the point cloud space. The x–y pixels coordinate of the binary image labeled as building facades are preserved as x–y coordinate of 3D point cloud (with open z value) labeled as building, and not considered in the remainder of our approach. Other points (negligible amount compare to the size of whole point cloud) are labeled as non-building class and will be later be classified as other classes e.g. car, tree, pedestrian and etc.

C. Voxel based segmentation

Although top view range image analysis generates a very fast segmentation result, there are a number of limitation to utilize it for the small vertical object such as pedestrian and cars. These limitations are overcome by using inner view (lateral) or ground based system in which, unlike top view the 3D data processing is done more precisely and the point view processing is closer to objects which provides a more detailed sampling of the objects. However, this leads to both advantages and disadvantages when processing the data. The

disadvantage of this method includes the demand for more processing power required to handle the increased volume of 3D data.

According to voxel based segmentation, points which are merely a consequence of a discrete sampling of 3D objects are merged into clusters voxels to represent enough discriminative features to label objects. 3D features such as intensity, area and normal angle are extracted based on these voxels. The voxel based classification method consists of three steps, voxelization of point cloud, merging of voxels into super-voxels and the supervised classification based on discriminative features extracted from super-voxels.

1) *Voxelization of Point Cloud:* In the voxelization step, an unorganized point cloud p is partitioned into small parts, called voxel v. The middle image in figure 4 illustrates an example of voxelization results, in which small vertical objects point cloud such as cars are broken into smaller partition. Different voxels are labelled with different colours. The aim of using voxelization is to reduce computation complexity by and to form a higher level representation of point cloud scene. Following [10], a number of points is grouped together to form a variable size voxels. The criteria of including a new point pin into an existing voxel i is essentially determined by the crucial minimal distance threshold d_{th} which is defined as equation (3).

$$\min(\|P_{im} - P_{in}\|_2) \le d_{th}, \quad 0 \le m, n \le N, \quad m \ne n \qquad (3)$$

Where p_{im} is an existing 3D point in voxel, p_{in} is a candidate point to merge to the voxel, i is the cluster index, d_{th} is the maximum distance between two point and N is the maximum point number of a cluster. If the condition is met, the new point is added and the process repeats until no more point that satisfies the condition is found. Equation (3) ensures that the distance between one point and its nearest neighbours belonging to the same cluster is less than d_{th}. Although the maximum voxel size is predefined, the actual voxel sizes depend on the maximum number of points in the voxel (N) and minimum distance between the neighbouring points.

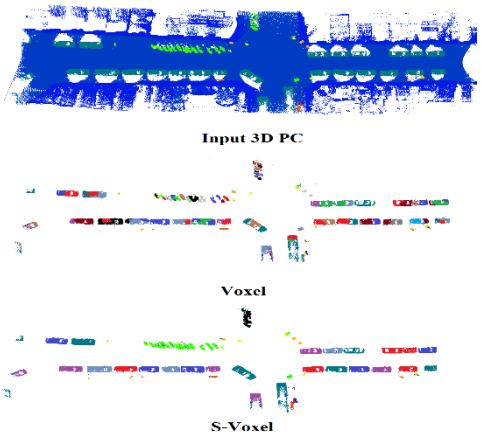

Fig 4. Voxelization of Point Cloud. from top to down: top view row point cloud, voxelization result of objects point cloud after removing ground and building, s-voxelization approach of point cloud

2) Super Voxelization: For transformation of a voxel to super voxel we propose an algorithm to merge voxels via region growing with respect to the following properties of clusters:

- If the *minimal geometrical distance*, D_{ij}, between two voxels is smaller than a given threshold, where D_{ij} is defined as equation (4):

$$D_{ij} = \min\left(\|P_{ik} - P_{jl}\|_2\right), k \in (1, m), l \in (1, n) \qquad (4)$$

  Where voxels v_i and v_j have m and n points respectively, and p_{ik} and p_{jl} are the 3D point belong to voxel v_i and v_j.
- If the *angle between normal vectors of two voxels* is smaller than a threshold: In this work, normal vector is calculated using PCA (Principal Component Analysis) [11]. The angle between two s-voxels is defined as angle between their normal vectors as equation 5:

$$\theta_{ij} = \arccos(< n_i, n_j >) \qquad (5)$$

  Where n_i and n_j are normal vectors at v_i and v_j respectively.

The proposed grouping algorithm merges the voxels by considering the geometrical distance ($D_{ij} < d_{th}$) and normal features of clusters ($\theta_{ij} < \theta_{th1}$). All these Voxelization steps then would be used in grouping these super-voxels (from now onwards referred to as s-voxels) into labeled objects.

The advantage of this approach is that we can now use the reduced number of s-voxels instead of using thousands of points in the dataset, to obtain similar results for classification.

3) *Feature extraction:* For each s-voxel, seven main features are extracted to train the classifier.

Geometrical shape: Projected bounding box has effective features due to the invariant dimension of objects. We extract four feature based on the projected bonding box to represent the geometry shape of objects.

-Area: the area of the bounding box is used for distinguishing large-scale objects and small ones.

-Edge ratio: the ratio of the long edge and short edge.

-Maximum edge: the maximum edge of bounding box.

-Covariance: is used to find relationships between points spreading along two largest edges.

Height above ground: Given a collection of 3D points with known geographic coordinates, the median height of all points is considered as the height feature of the s-voxel. The height information is independent of camera pose and is calculated by measuring the distance between points and the road ground.

Horizontal distance to center line of street: Following [12], we compute the horizontal distance of the each s-voxel to the center line of street as second geographical feature. The street line is estimated by fitting a quadratic curve to the segmented ground.

Density: Some objects with porous structure such as fence and car with windows, have lower density of point cloud as compared to others such as trees and vegetation. Therefore,

the number of 3D points in a s-voxel is used as a strong cue to distinguish different classes.

Intensity: following [12], LiDAR systems provide not only positioning information but also reflectance property, referred to as intensity, of laser scanned objects. This intensity feature is used in our system, in combination with other features, to classify 3D points. More specifically, the median intensity of points in each s-voxel is used to train the classifier.

Normal angle: Following [13], we adopt a more accurate method to compute the surface normal by fitting a plane to the 3D points in each s-voxel. The surface normal is important properties of a geometric surface, and is frequently used to determine the orientation and general shape of objects.

Planarity: Patch planarity is defined as the average square distance of all 3D points from the best fitted plane computed by RANSAC algorithm. This feature is useful for distinguishing planar objects with smooth surface like cars form non planar ones such as trees.

4) Classifier: The Boosted decision tree [14] has demonstrated superior classification accuracy and robustness in many multi-class classification tasks. Acting as weaker learners, decision trees automatically select features that are relevant to the given classification problem. Given different weights of training samples, multiple trees are trained to minimize average classification errors. Subsequently, boosting is done by logistic regression version of Adaboost to achieve higher accuracy with multiple trees combined together. Each decision tree provides a partitioning of the data and outputs a confidence-weighted decision which is the class-conditional log-likelihood ratio for the current weighted distribution. In our experiments, we boost 10 decision trees each of which has 6 leaf nodes.

D. 2D-3D association

With the advancement of LiDAR sensors, GPS and IMU devices, large-scale, accurate and dense point cloud can be created and used for 2D scene parsing purpose. The cubic images and 3D labeled LiDAR point are the inputs of the processing step and parsing results are image segments assigned with different class labels. The proposed parsing pipeline aligns 3D LiDAR point cloud with 2D images. Input images are segmented into superpixels to reduce computational complexity and to maintain sharp class boundaries. Each superpixel in 2D image is associated with a collection of labeled LiDAR points, which is assumed to form a planar patch in 3D world (fig 5).

Labeled images are generated at cubic image locations. All labeled LiDAR points are converted into local coordinates cantered at the panoramic image locations and then mapped onto the superpixels in the four cube faces. If multiple points fall into the same superpixel, the point with minimum distance to the image location is chosen to represent the label of superpixel. In the other words the label of 3D point which has the minimum distance to the image location along whole other 3D patch points, assumed as image superpixel label.

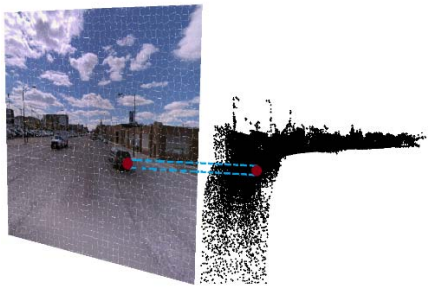

Figure 5. From 3D world to image plane

III. EXPERIMENTAL RESULT

Since no labeled image dataset consisting of corresponding LiDAR point cloud was available, we created and used labeled dataset of driving sequence from NAVTAQ True, provided by HERE, consists of best of sensors in three categories - positioning sensors, LiDAR sensors and imaging sensors. 10 semantic object classes are defined to label the image and corresponding LiDAR dataset: building, tree, sky, car, sign symbol, person, ground, fence, sidewalk and water. It's noteworthy that several objects such as wall sign and wall light are considered as building facades. The whole NAVTAQ True datasets are divided into two portions: the training set, and the testing set. The 70% long of dataset are randomly selected and mixed for training of classifier and 30% remained long of point cloud is used for testing.

Table 1. Confusion matrix of NAVTAQ True Database

	Sky	Building	Road	Tree	Car	Sidewalk	Sign- S	Fence	person	Water
Sky	96	2	0	2	0	0	0	0	0	0
Building	4	90	0	3	0	2	0	1	0	0
Road	2	0	96	0	1	1	0	0	0	0
Tree	6	17	0	74	0	3	0	0	0	0
Car	5	11	35	1	35	11	0	2	0	0
Sidewalk	2	4	12	1	4	77	0	0	0	0
Sign- S	8	2	5	4	3	60	17	0	0	1
Fence	5	37	0	3	4	1	0	49	0	1
person	10	34	1	3	3	21	0	0	22	6
Water	48	6	1	5	1	5	0	1	0	33

Mixing data from different cities poses serious challenges to the parsing pipeline, which is reflected by the decrease in the class average accuracy. Nevertheless, it seems our algorithm performs well on most per class accuracies, with the highest accuracy 96% achieved for the sky and ground and the lowest 17% for sign. This low accuracy for small objects (e.g. person, sign) is mainly due to lack of sufficient training examples, which naturally lead to a less statistically significant labeling for objects in these classes. Moving objects are even harder to reconstruct based solely on 3D data. As these objects (typically vehicles, people) are moving

through the scene, which make them appear like a long-drawn shadow in the registered point cloud. The global accuracy is about 88 %.

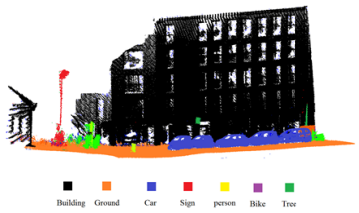

Figure 6. 3D Scene object recognition qualitative results in different view (point cloud)

As it can be seen in the figure 6, successful point cloud classification and alignment have been done accurately.

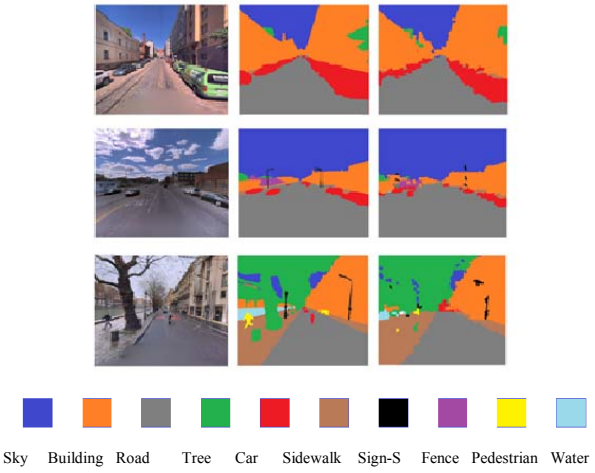

Figure 7. Scene parsing qualitative results. (Left to right): test image, ground (manually labeled image), parsing result

To compare our experimental result with other publications we also test Velodyne LiDAR dataset which only contains 3D point cloud in local coordinate system and there is not any image to test our 2D image parsing algorithm with. We train our classifier using seven scene datasets, selected randomly, and test on the remaining three scenes. Our algorithm performs well on most per class accuracies with the heights accuracy 98% for ground and the lowest 72% for sign-symbol. The global accuracy and per-class accuracy are about 94% and 87% respectively. We also compare our approach to the method described by Lai in [15]. Table 2 shows its quantitative testing result. In terms of per class accuracy, we achieve 87% in comparison to 76%.

Table 2. Comparison of the class accuracy of our approach and Lais approach

	Tree	Car	Sign	person	Fence	Ground	Building
Lai	0.83	0.91	0.80	0.41	0.61	0.94	0.86
Our	0.89	0.95	0.72	0.88	0.85	0.98	0.95

IV. CONCLUSION

We have proposed a novel and comprehensive framework for semantic parsing of street view. Using unsupervised segmentation huge amount of data (more than 75% of points) are labeled, and only small amount of point cloud which have complex shape remained to be segmented. The proposed two-stage method requires only small amount of time for training while the classification accuracy is robust to different types of LiDAR point clouds acquisition methods.

REFERENCES

[1] C. Liu, J. Yuen, Torralba, nonparametric scene parsing: Label transfer via dense scene alignment. In: Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on, IEEE 2009.

[2] G. Csurka, F. Perronnin, A simple high performance approach to semantic segmentation. In: BMVC, 2008.

[3] G. Floros, B. Leibe, Joint 2d-3d temporally consistent semantic segmentation of street scenes. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2823-2830.

[4] G. Zhang, J. Jia, T. Wong, H. Bao, Consistent depth maps recovery from a video sequence, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (2009) 974-988.

[5] W. L Lu, K. P. Murphy, and J. J. Little, A. Shefier, H. Fu, A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data, Geoscience and Remote Sensing, IEEE Transactions on 47 (2009) 2913-2922.

[6] J, Hern_andez, B. Marcotegui, Filtering of artifacts and pavement segmentation from mobile lidar data, In: ISPRS Workshop Laserscanning 2009.

[7] Y. Zhou, Y. Yu, G. Lu, S. Du, Super-segments based classification of 3D urban street scenes. Int J Adv Robotic Sy 9, 2012.

[8] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395, 1981.

[9] T. Pavlidis, Algorithms for graphics and image processing. Computer science press, 1982.

[10] Y. Zhou, Y. Yu, G. Lu, S. Du, Super-segments based classification of 3d urban street scenes. Int J Adv Robotic Sy 9 (2012)

[11] K. Klasing, D. Althoff, D. Wollherr, M. Buss, Comparison of surface normal estimation methods for range sensing applications. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE (2009) 3206–3211, 2009

[12] P. Babahajiani, L. Fan, M. Gabbouj, Semantic parsing of street scene images using 3D lidar point cloud. Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops 13, 2013.

[13] J. Xiao, L. Quan, Multiple view semantic segmentation for street view images. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE 2009.

[14] M. Collins, R. E. Schapire, Y. Singer, Logistic regression, adaboost and bregman distances. Machine Learning 48, 2002.

[15] K. Lai, D. Fox, Object recognition in 3d point clouds using web data and domain adaptation. The International Journal of Robotics Research 29, 2010.