

# Automatic detection of pole-like structures in 3D urban environments

Federico Tombari<sup>1</sup>  
Samuele Salti<sup>1</sup>

Nicola Fioraio<sup>1</sup>  
Alioscia Petrelli<sup>1</sup>

Tommaso Cavallari<sup>1</sup>  
Luigi Di Stefano<sup>1</sup>

**Abstract**—This work aims at automatic detection of man-made pole-like structures in scans of urban environments acquired by a 3D sensor mounted on top a moving vehicle. Pole-like structures, such as e.g. road signs and streetlights, are widespread in these environments, and their reliable detection is relevant to applications dealing with autonomous navigation, facility damage detection, city planning and maintenance. Yet, due to the characteristic thin shape, detection of man-made pole-like structures is significantly prone to both noise as well as occlusions and clutter, the latter being pervasive nuisances when scanning urban environments. Our approach is based on a “local” stage, whereby local features are classified and clustered together, followed by a “global” stage aimed at further classification of candidate entities. The proposed pipeline turns out effective in experiments on a standard publicly available dataset as well as on a challenging dataset acquired during the project for validation purposes.

## I. INTRODUCTION AND RELATED WORK

Advanced technologies for automatic processing of 3D urban data are increasingly demanded by the market due to the recent widespread of 3D sensors that can be easily mounted on top of a roaming vehicle. Indeed, in the past few years these sensors, such as the Velodyne, have become more and more affordable, and allow acquiring dense and accurate 360° scans of the environment surrounding the vehicle at high frame rates (e.g. 100 Hz). They are usually paired with a Inertial Measurement Unit (IMU) and a GPS system so to automatically associate each scan with vehicle trajectory coordinates and geo-locate each acquired 3D point. Typical applications exploiting these sensing modalities deal with road and facility management, so as to detect damages and faults on roads, road facilities and buildings, as well as autonomous driving.

One of the most common and relevant class of objects that can be found in 3D urban scans is represented by man-made pole-like structures (road signs, streetlights, utility poles, traffic lights, ...). Indeed, reliable detection of these structures is mandatory for both the aforementioned applications: in particular, road signs and traffic lights for autonomous driving, all pole-like structures for facility damage detection and road maintenance. Yet, manual annotation is cumbersome and time wasting due to both the huge amount of data typically acquired by roaming vehicles (of the order of thousands of points per square meter, which typically implies tens of millions of 3D points per each kilometer of trajectory) as well as the challenging objects that need to be localized, often represented by thin and significantly

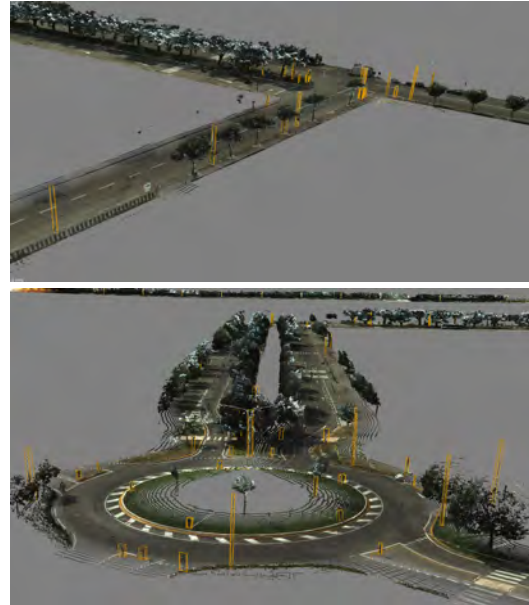


Fig. 1: Detection of relevant pole-like structures in urban environments acquired by a 3D laser scanner mounted on a roaming vehicle is a challenging task. The orange bounding boxes denote the structures detected automatically by our algorithm.

occluded structures. Fig. 1 shows two examples of a typical scenario of the kind, with the 3D orange bounding boxes highlighting the fully automatic detections provided by the method proposed in this paper. As it can be observed, the automatic detection task is very hard due to the presence of vegetation often surrounding pole-like structures, the extremely thin shape of the sought objects and the similar shape of tree trunks.

The above considerations motivate research and development of techniques for automatic detection of pole-like structures in 3D data. Early work on the subject [1] relies on a set of heuristics on range data so to extract possible pole-like structures. Upon certain conditions (poles higher than the road level), successive manual annotation is required. Other approaches regard the joint use of 3D data and intensity images. In [2], a multi-sensor setup is deployed to improve automatic recognition of road facilities such as road signs, traffic marks and road fences. In particular, the authors present a method exploiting an inertial system, an odometer and two GPS receivers to accurately align the intensity data acquired from two sets of stereo cameras and the range data

<sup>1</sup>The authors are with DISI, University of Bologna, 40136 Bologna, Italy  
{name.surname}@unibo.it



Fig. 2: Proposed pipeline for automatic pole detection within 3D data.

acquired from three laser scanners, all mounted on-board a vehicle. Extraction of pole-like structures such as road signs can then be improved by cross-checking road sign detection in 2D data with pole detection in 3D data. The work in [3] proposes to detect street lamps via analysis of 2D data only, i.e. based on images acquired by 6 partially overlapping cameras installed on-board a vehicle. First, detection of vertical structures is carried out by SVM classification of a feature vector relying on image edges. Next, pole hypotheses are cast in the 2D space represented by the road plane by exploiting the knowledge of extrinsic parameters of the cameras: local maxima within this space identify possible poles seen by different cameras. The lack of 3D data, though, yields spatially imprecise localization (e.g. with an accuracy of about 3 meters).

More recent work focused on analysis of 3D data usually deploys principal directions computed from the scatter matrix of a set of points to derive the “poleness” of a 3D region. The work in [4] aims at detection of pole-like structures (including tree trunks) based on the analysis of laser scanner data. First, a 1-NN segmentation of laser scanlines is performed to yield spatially compact 1D clusters along scanlines. Then, similar 1D clusters on adjacent scanlines are merged iteratively. Non-adjacent clusters are then merged by evaluating the distance from the vertical principal direction of each cluster. Finally, clusters are classified as poles by checking five heuristic conditions relying on fixed thresholds. [5] proposes a pole extraction and classification algorithm based on k-Nearest Neighbor (kNN) segmentation and evaluation of the principal directions of each segmented cluster. The use of an initial blind segmentation step seems to hinder performance notably, as the reported classification accuracy is 97.4% for pole segments and 63.9% for all segments. In [6], an algorithm for extraction of pole-like structures from Lidar data is proposed. Unlike [5], principal directions are determined from the covariance matrix computed on a neighborhood of each point. Eigenvalues are then thresholded to label each point as either belonging to a pole or not. Region growing is applied to cluster the point cloud into pole hypotheses, and the pole radius is estimated via fitting a cylinder to each cluster. Reported classification accuracies on 2 datasets proposed by the authors are 33% and 91%. The approach described in [7] projects the 3D point cloud data into 2D elevation images called *digital elevation models*. The 2D images are then elaborated with standard image processing tools (e.g. morphological operators, watershed transform, ...) in order to segment the objects above the ground level. The objects are then classified using a Support Vector Machine. The segmentation recall rates for the “pole-like” categories (“lamppost”, “light pole” and “traffic

light”) are respectively 82%, 77% and 81%. The successive classification step reports a 88% precision and 85% recall for the “pole-like” object category as a whole.

In this paper, we present a novel method for pole-like structure detection in 3D data. One main novel contribution of our work concerns the deployment of specifically devised 3D descriptors to be computed at both point level (*local* descriptors) as well as cluster level (*global* descriptors), which hold the potential to rely on more discriminative cues than, e.g., principal directions. The method consists in a pipeline of several stages. First, 3D data are processed at point level, with the goal of clustering 3D points together when they form meaningful pole-like structure: this has the advantage of providing a more meaningful clustering of 3D points with respect to, e.g., [5]. Then, further processing occurs at cluster level, so to discard wrong pole-like segments which would otherwise lead to false positives. We show experimental results on a publicly available urban data benchmark dataset, as well as on our own dataset acquired via a laser scanner mounted on top a moving vehicle.

## II. A 3D PIPELINE FOR AUTOMATIC POLE DETECTION

As anticipated in Section I, the proposed 3D pole extraction approach deploys a pipeline encompassing several stages, as sketched in Fig. 2. The pipeline input is a 3D point cloud acquired by a 3D sensor such as a range laser scanner. Moreover, we estimate a normal vector at each 3D point as the Eigen vector associated to the smallest Eigen value of the covariance matrix, approximated by computing the scatter matrix from all the neighbors at a given distance. Instead of generating an optimal radius for each point as proposed in [8], we got fairly good results by simply fixing it at a reasonable value.

During the initial pre-processing step, planar structures are detected and removed by means of a RANSAC-like iterative algorithm. At this stage, not only the main road plane is removed, but also other planar surfaces, such as building façades and ground regions, so as to both reduce the size of the cloud to be processed for the sake of computational efficiency as well as get rid soon of many regions which are guaranteed not to correspond to the sought pole-like structures

Next, on each point not discarded by the initial plane-removal step, a *local* feature specifically designed to highlight the *poleness* of a point’s neighborhood is computed and then classified by means of a Support Vector Machine (SVM). Following *local* classification, a semantic clustering relying on a Markov Random Field (MRF) formulation over the connected point cloud graph is carried out [9]. By exploiting both point connectivity as well as the continuous



Fig. 3: Road pavement detection and removal (red points). The point clouds come from the Oakland 3-D Point Cloud Dataset (left) and our urban dataset (right); for details see Sec. III.

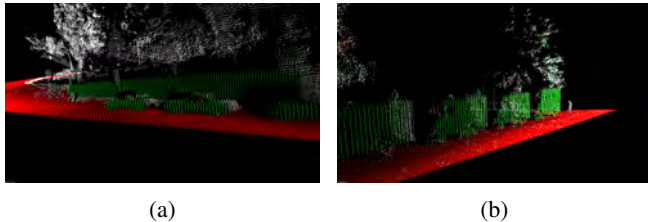


Fig. 4: Non-horizontal (green) and road planes (red) extraction in two point clouds from the Oakland 3-D Point Cloud Dataset (see Sec. III); notice how parked vehicles, which are distractors as far as pole detection is concerned, are removed as non-horizontal planes.

output of the classifier, this step subdivides the cloud into clusters, each labeled as either being a “pole” or not.

Although MRF-based clustering would already provide the required type of output, we have devised a more elaborate approach whereby additional features are extracted at cluster level (i.e., one for each cluster) and then classified. This additional step turns out effective in improving significantly the classification accuracy of the whole system by reducing the number of both false positives and missed poles. Each stage is described in detail in the remainder of this Section.

#### A. Preprocessing: plane removal

Planes are detected and removed by a RANSAC-based plane model fitting. However, though the upward direction is already provided by the IMU, more often than not the main road plane cannot be represented by a single plane model due to road discontinuities, noise and slope variations. Indeed, the road is usually made-out of at least two planes (i.e. one for each lane). Therefore, we detect the planes associated with the road by iterative plane extraction:

- 1) find the current dominant plane (as described soon later);
- 2) if the number of extracted points is below a threshold  $\Rightarrow$  exit;
- 3) if the absolute value of the scalar product between the plane normal and the upward direction is above a threshold  $\Rightarrow$  the plane is labeled as *road*;
- 4) remove the points belonging to the found plane from the cloud and repeat.

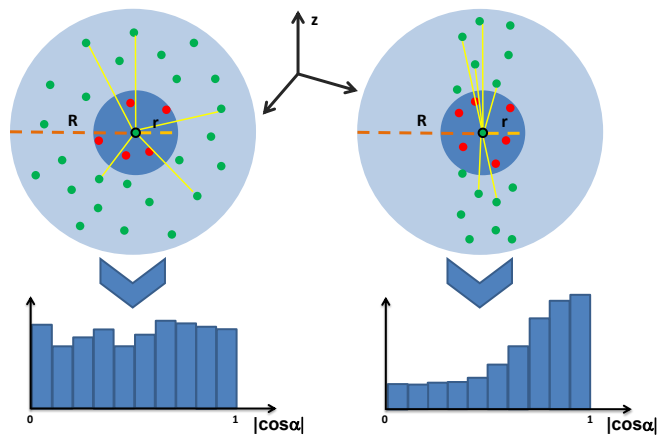


Fig. 5: The proposed local feature for pole detection entails a distinctive shape in presence of pole-like objects.

Instead of using the standard RANSAC algorithm, which would require at least 3 points for plane fitting [10], we propose to find the dominant plane by means of a 1-point RANSAC plane fitting. For any randomly sampled point  $\mathbf{p}$  with normal  $\mathbf{n}_p$ , first we consider the plane  $\mathbf{n}_p \cdot (\mathbf{x} - \mathbf{p}) = 0$ . Then, a point  $\mathbf{q}$  with normal  $\mathbf{n}_q$  is assigned to the consensus set if the absolute value of the scalar product between  $\mathbf{n}_p$  and  $\mathbf{n}_q$  is above a threshold and the distance from the plane projected along  $\mathbf{n}_q$  is below a given value. The plane model with the largest consensus is kept as the dominant plane. Thus, while the standard approach may be confused by randomly aligned points, our estimation of the RANSAC plane model from point normals is robust and reliable, as a normal vector implicitly refers to a plane which locally fits a point’s neighborhood. The employed RANSAC algorithm does not guarantee compactness of the found planes, therefore an Euclidean clustering algorithm is run on the points labeled as *road* and, accordingly, clusters which turn out too small are discarded.

In our approach, we apply the same algorithm to detect and remove non-horizontal planes, such as those belonging to façades and walls. In this case, of course, at algorithm step 3 the absolute value of the scalar product between the plane normal and the upward direction must turn out *below* a threshold. Some results concerning the plane removal algorithm are shown in Fig. 3 and 4.

#### B. Local feature extraction and classification

In this stage we rely on a novel feature descriptor specifically designed to highlight pole-like structures, which is computed densely on the points not classified as planes by the preprocessing phase. As illustrated in Fig. 5, given one such point  $\mathbf{p}$ , first we collect all its neighbors falling between two spheres of radii  $r$  and  $R$  centered at  $\mathbf{p}$ , with  $r < R$  (the light blue region in Fig. 5); then, for each valid neighbor  $\mathbf{q}$ , we accumulate the absolute value of the scalar product between  $(\mathbf{p} - \mathbf{q})$  and the known upward direction into a histogram of  $N$  bins. Accordingly, points belonging to pole-like structures are expected to exhibit a distinctive histogram

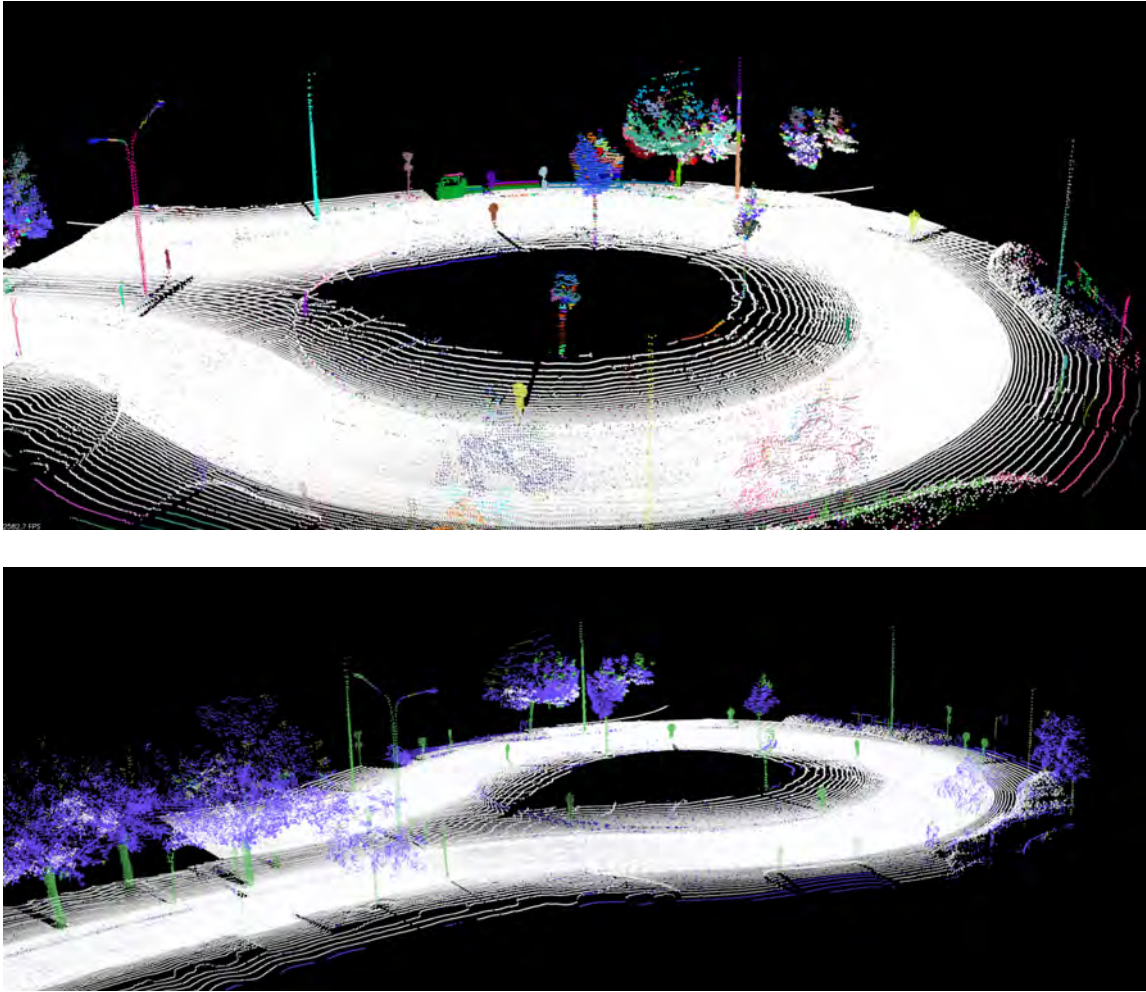


Fig. 6: The MRF-based semantic clustering stage improves the point-wise SVM classification by determining 3D clusters with an associated semantic label. Top: clustered connected components (shown with pseudo-colors); Bottom: assigned semantic labels (green: “pole”, violet: “not-pole”).

with a few high-count bins toward the right end, unlike other “non-pole” regions (see Fig. 5).

Given a sufficient number of examples labeled as “pole” or “not-pole”, we train a SVM classifier to distinguish between the two categories. However, the point-wise classification provided by the SVM often leads to a noisy output with spurious labels within a group of correctly classified points. Therefore, instead of looking for a hard classification threshold, we use the classifier to obtain a score related to the probability of a point to either belonging to a pole or not, leaving the final decision to the successive semantic clustering step, which combines the local independent observations provided by the SVM together with spatial regularization constraints.

### C. MRF-based semantic clustering

The previous stage associates to each 3D point a continuous value denoting how well the point’s neighborhood can be assimilated to a pole-like structure. This information is exploited at this stage with the goal of carrying out semantic clustering, i.e. subdividing the point cloud into clusters of points, each labeled either as “pole” or “not-pole”. This is

useful not only to improve the classification accuracy by spatially enforcing smoothness in the labeling process, but also to apply a successive segment-wise classification, as explained in Section II-D.

Purposely, we have employed the MRF formulation proposed in [9]. Accordingly, a connected graph is built on top the point cloud by associating each point to its  $k$  nearest neighbors ( $k = 4$  has been used in our experiments). Then, the following energy functional defined over the graph is minimized:

$$E(X) = \sum_i \phi_i(x_i) + \sum_i \phi_{i,j}(x_i, x_j) \quad (1)$$

where  $x_i$  is the label assigned to node  $i$  and  $j$  is one of its neighbors according to the connectivity established by the graph. This functional can be interpreted as the *energy* associated with a graph configuration,  $X$ , and is composed of two terms; the *unary* term,  $\phi_i(x_i)$ , aims at maximizing the coherence with the observations provided by the SVM classifier, the *pairwise* term,  $\phi_{i,j}(x_i, x_j)$ , enforces label consistency between neighboring points.

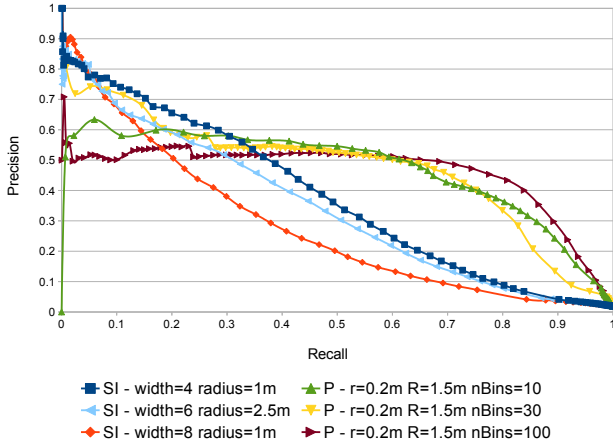


Fig. 7: Point-wise classification on the *CMU-Oakland* dataset: proposed “poleness” feature (P) vs. Spin Images (SI). For both methods, the 3 best parameter choices are reported in the chart.

As in [9], an approximated solution,  $\tilde{X}$ , is obtained via Loopy Belief Propagation [11]. Given  $\tilde{X}$ , a simple connected component algorithm applied over the 3D graph can segment the data into 3D clusters with associated semantic labels. In particular, in our implementation we have used an iterative *breadth-first search* algorithm (BFS) [12]. Fig. 6 provides an example of the output from the semantic clustering stage of our pipeline, showing both unlabeled connected components (top) as well as clusters semantically labeled as either “pole” or “not-pole” (bottom).

#### D. Global feature extraction and classification

Although the algorithm described so far is able to detect reliably most pole-like structures, it generally yields a high number of false positives. Specifically, tall and thin structures, such as tree trunks and parts of buildings or walls not completely discarded by the plane removal step, may be returned as *pole* candidates. In order to prune away as many false positives as possible, we enforce an additional stage aimed at classifying entities at cluster rather than point level.

In particular, each cluster yielded by the previous MRF-based stage is described via a *global* feature and classified by means of a SVM classifier. To this purpose, a proper cluster descriptor has to be employed. After a careful evaluation, we have selected the well known Spin Images descriptor [13], and devised an adaptation to cluster representation which enables a more distinctive description of pole-like structures. In particular, at each cluster we compute a Spin Image using, as center, the geometric centroid of the cluster itself and, as the radius defining the extension of the description support, the maximum Euclidean distance between the centroid and all points belonging to the cluster, this allowing all cluster points to be included into the description. The Spin Images algorithm deploys the normal of the point for which the

descriptor is being computed as the Rotation Axis around which a 2D plane spins to accumulate all points into a histogram. To get improved robustness to noise, we propose to use a different Rotation Axis: first we compute the covariance matrix of the coordinates of the points belonging to the cluster point, then use as Rotation Axis the eigenvector corresponding to the largest eigenvalue. To define a repeatable sign for the eigenvector, we always choose the direction that points “upwards” wrt. the road ground plane, thus exploiting the presence of the Global Reference Frame of the point cloud. The use of a Rotation Axis defined on the cluster as a whole establishes a reference direction more robustly than the standard Spin Images approach, as local computation of the normal may turn out fragile in case of thin objects and noisy measurements.

The feature just described is not sufficient to accurately reject all false positive instances: in a few cases tree trunks are incorrectly classified as poles, due to their shape, indeed, closely resembling a man-made pole. Clearly, the semantic difference between a pole and a trunk does not lie in its vertical section but in the points located in the area just *outside* the cluster: typically foliage is present around the top of a tree trunk whereas a utility pole or road sign does not include such characteristic trait. For this reason, we propose to associate to each cluster a description of its context too, which is obtained by means of a second Spin Image descriptor computed using the same centroid, support radius and rotation axis as before. However, as for this second Spin Image capturing the context of a cluster, the description relies not just on the points belonging to the cluster, but instead also on the original input point cloud, so as to include contributions from nearby structures (road plane, façades, foliage...).

The final *context-aware* global descriptor used for the SVM classification at cluster level is obtained by concatenation of the two Spin Images.

### III. EXPERIMENTAL RESULTS

Qualitative and quantitative experiments have been conducted to evaluate each of the described stages, as well as the overall pole detection pipeline. Purposely, we have used two different datasets, both concerning an urban scenario acquired by a laser scanner. The first dataset is the public *CMU-Oakland* Point Cloud Dataset<sup>1</sup> [14], which includes 17 point clouds with point-wise ground-truth labels related to semantic categories. However, as we aim at pole detection, the available labels have been turned into the pair “pole” and “not-pole”. As the “pole” ground-truth category includes both utility poles and tree trunks, to evaluate the proposed approach at cluster level (Sec. III-B) we have generated additional ground-truth information by drawing into the point clouds 3D bounding boxes around clusters and manually tagging each as either “pole” or “not-pole”.

The second dataset, hereinafter referred to as *City*, comprises two rather long sequences that we have acquired by

<sup>1</sup>available at [http://www.cs.cmu.edu/~vmr/datasets/oakland\\_3d/cvpr09/doc/](http://www.cs.cmu.edu/~vmr/datasets/oakland_3d/cvpr09/doc/)

TABLE I: Improvement in point-wise classification due to the MRF-based clustering stage on the *CMU-Oakland* dataset

	TP	FP	Precision	Recall
<i>before</i> -MRF	3427	29995	0.103	0.958
<i>after</i> -MRF	3444	27928	0.110	0.963

an off-the-shelf system based on a laser scanner mounted on a moving vehicle. The acquisition device is engineered so as to align automatically the 3D data based on GPS and IMU measurements, thus providing huge point clouds that we then cut and process in smaller chunks for the sake of computational feasibility. As described for the *CMU-Oakland* dataset, we have generated cluster-level ground-truth also for *City* by manually tagging pole-like structures.

#### A. Local classification and clustering

We have evaluated the proposed local feature highlighting the “poleness” of a neighbourhood by comparing it to the well-known Spin Image descriptor [13] on the *CMU-Oakland* dataset<sup>2</sup>. We tuned the parameters of both features via cross-validation, in particular by varying the internal ( $r$ ) and external ( $R$ ) radius and number of bins for our proposal, the image width  $W$  and cylinder radius  $b$  for Spin Images[13]. Between the available clouds, 12 were chosen for training and cross-validation, while the remaining 5 were employed for testing. The results plotted in Fig. 7 show clearly the superior performance delivered by our proposal with respect to Spin Images on a wide range of classification thresholds, the latter turning out too imprecise - unless one accepts very low sensitivity - due to the difficulty of estimating reliably the normal direction over thin pole-like structures. Indeed, we found the scatter matrix computed at points lying on these objects to provide quite often two small and similar eigenvalues, which results in ambiguous estimation of the normal direction.

In addition to providing the successive global classification stage with meaningful pole candidates, we would expect MRF-based semantic clustering to be able to correct some of the errors yielded by the point-wise SVM classification relying on the “poleness” feature. Accordingly, in Table I we compare the number of TPs and FPs *before* and *after* the MRF-based semantic clustering stage. More precisely, using the same test set as for the evaluation of point-wise classification and considering the probability score provided by the SVM classifier, we compare the points for which  $\Pr(\text{“pole”}) > \Pr(\text{“not-pole”})$  (*before*-MRF classification) to those labeled as “pole” by the MRF-based step (*after*-MRF classification). As shown in the Table, on a total of 3577 “pole” points we got a sensible reduction of false positives together with a slight increase of the number of true positives. At the same time, and as previously explained, the MRF-based stage guarantees homogeneously connected labels, as required for the successive global classification stage.

<sup>2</sup>Point-wise ground-truth is not available for the *City* dataset

Finally, as already pointed out, an example of the data inputted to the final global classification stage is depicted in Fig. 6. We wish to point out that, although the MRF stage is able to reliably eliminate many false positive points as witnessed by Table I, to retain the majority of true poles present in the scene the number of false positive clusters is still high: in particular, the number of TP/FP clusters is, respectively, 32/256 for the *CMU-Oakland* dataset and 84/859 for the *City* dataset. This motivates the use of a successive global classification stage, which is evaluated in the next subsection.

#### B. Global classification

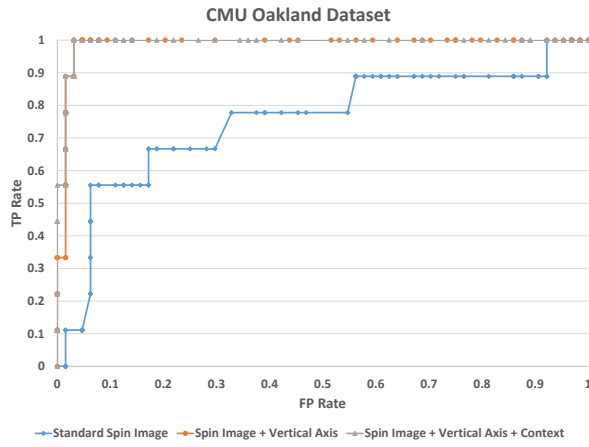
To evaluate the performance of the proposed global classification stage we used both the previously mentioned datasets, i.e. *CMU-Oakland* and *City*. The latter has been split into two parts, one used for evaluation of the global classifier and the other to assess the performance of the pipeline as a whole (see Subsection III-C). Specifically, in each of the two datasets 70% of the clusters were randomly selected to become the training set, the remaining 30% used for testing. We used SVM with a Radial Basis Function kernel and ran a 10-fold cross validation process to find the optimal  $C$  and  $\gamma$  parameters.

The classification performance was evaluated using three different cluster descriptors: a standard Spin Image (with parameter  $W = 8$ ), a Spin Image based on the different choice of the reference direction (Section II-D) and a context-aware Spin Image (see again Section II-D). Unlike the two variants described in Section II-D, the standard SI descriptor is computed using as center the cluster point closest to the centroid and taking the normal at this point as rotation axis, while, similarly to the variants in Section II-D, the extent of the description support (i.e. the cylinder radius) is given by the Euclidean distance from the center to the cluster point farthest away from the center.

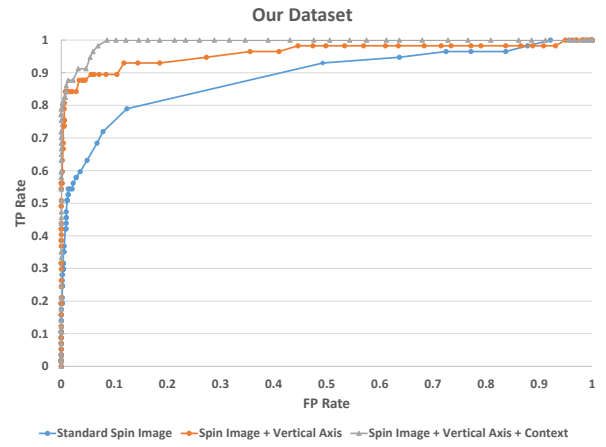
As for the evaluation, in Figure 8 we report the ROC curves computed by varying the SVM decision threshold related to binary classification of the candidate clusters coming from the MRF-stage into either “pole” or “not-pole”. Both Figure 8a and 8b show that the choice as reference direction of the eigenvector corresponding to the largest eigenvalue of a cluster improves significantly the classification performance with respect to the standard Spin Image descriptor. Due to the much smaller number of pole-like structures present in *CMU-Oakland*, the benefits brought in by deployment of context information are evident only in the ROC relative to the *City* dataset (Figure 8b): here the high number of trees and other distractors would increase considerably the false positive rate unless the cluster descriptor is endowed with context information.

#### C. Overall evaluation

In the previous sections we presented results focused on evaluating individually each stage of the proposed pipeline. In this Subsection, instead, we evaluate the entire algorithm as a whole. To this aim, we use a subset of the *City* dataset

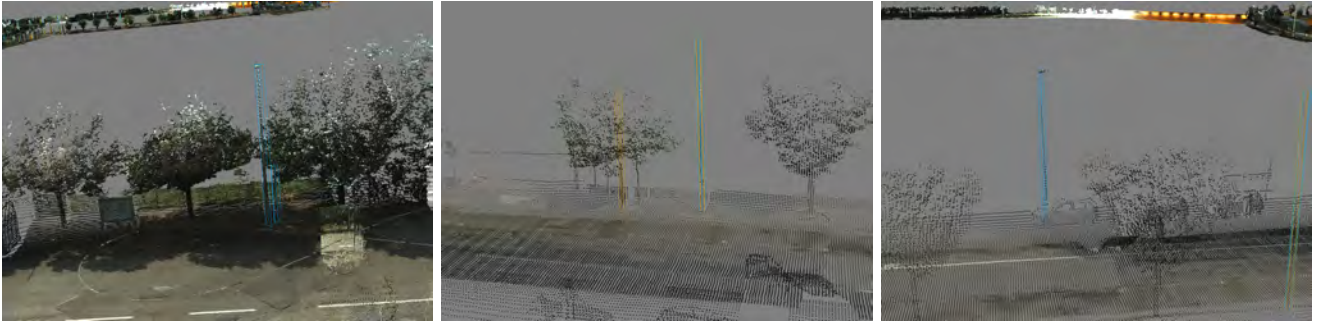


(a) *CMU-Oakland* dataset: the choice of the largest cluster eigenvector as reference direction improves performance with respect to standard Spin Images, while the small number of poles prevents evaluation of the impact of the context-aware descriptor.



(b) *City* dataset: deployment of context allows for a conspicuous reduction of the FP rate at high TP rates.

Fig. 8: Quantitative evaluation of the global classification stage.



(a) Missed poles due to the clutter caused by (b) A tree is incorrectly classified as a pole. (c) Another false negative due to the small amount of 3D points captured by the sensor.

Fig. 9: Sample results yielded by the proposed pipeline. The orange bounding boxes are the detections provided by our algorithm while the blue boxes represent the ground truth.

composed of 385 point clouds that were used neither for training nor for evaluation of the individual pipeline stages. Given the manually annotated ground-truth concerning each pole, the proposed algorithm was applied on the test point clouds. Successful detections were determined by comparing each detected bounding box with the closest ground truth bounding box, in particular we require a minimum overlap of 70%, carried out by checking the following condition:

$$\frac{V_{ratio} + V_{overlap\_ratio}}{2} \geq 0.7 \quad (2)$$

$$V_{ratio} = \frac{\min(V_d, V_{gt})}{\max(V_d, V_{gt})}$$

$$V_{overlap\_ratio} = \frac{V_{overlap}}{\min(V_d, V_{gt})}$$

where  $V_d$  is the volume of the detected bounding box,  $V_{gt}$  the volume of the ground truth bounding box,  $V_{overlap}$  the volume of the intersection between the two boxes.

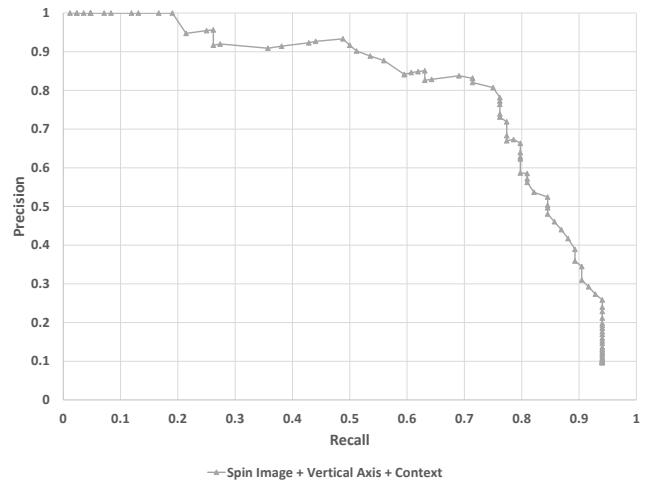


Fig. 10: Overall performance of the proposed algorithm on the *City* dataset. The Recall saturates at 94% mainly because of misdetections due to vegetation hiding poles.

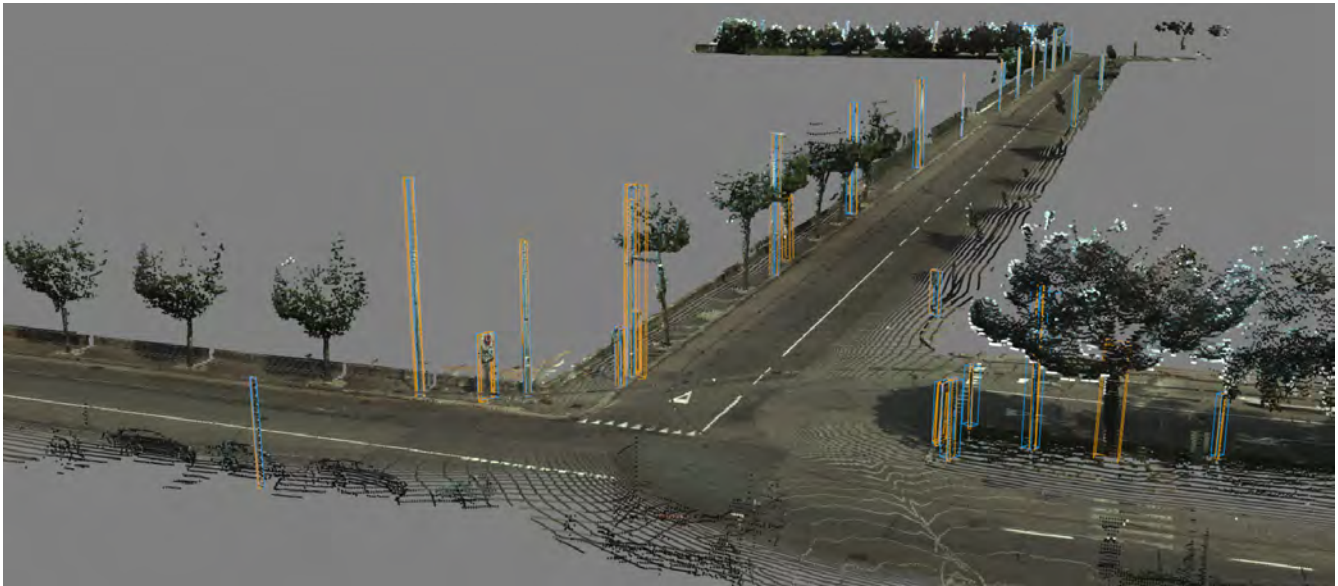


Fig. 11: Cluttered area with many diverse pole-like structures, such as streetlights and traffic signs, detected correctly and a few errors.

Fig. 10 shows the *Precision-Recall* curve yielded by the overall pipeline on the 385 test point clouds of the *City* dataset, where, similarly to Figure 8, the curve spans the decision threshold of the final global classifier. For example, a possible operating point of the curve would deliver 75% recall rate with 80.7% precision; alternatively, should the application demand higher recall, one might reach 85% recall with 52.5% precision rate or 89% recall with 39% precision.

Finally, Figures 1, 9 and 11 report some qualitative output yielded by the proposed pole detection pipeline, which typically provide only a few false positives and negatives even in presence of heavy clutter and poles featuring diverse form factors. Additional qualitative results related to the proposed pipeline are available in the accompanying video.

#### IV. CONCLUSIONS

We have proposed an algorithm for automatic detection of pole-like structures in 3D scans of urban environments. The overall pipeline builds upon four main stages: removal of planar surfaces, local classification by a “poleness” descriptor, MRF-based clustering and final context-aware cluster classification. The proposed approach can detect effectively pole-like structures within highly cluttered environments, with the additional ability of discriminating between tree trunks and poles due to the deployment of contextual information. This allows to reach performance levels such as 75% Recall and 80.7% Precision on a challenging dataset acquired for validation purposes. The major sources of misdetections in these kind of data turn out poles heavily occluded by vegetation or made out of just a few 3D points.

#### ACKNOWLEDGMENTS

We wish to thank QONSULT SpA for funding the research activity leading to the results presented in this paper.

#### REFERENCES

- [1] D. Manandhar and R. Shibasaki, “Feature extraction from range data,” in *Proc. Asian Conf. on Remote Sensing*, 2001.
- [2] S. Shibasaki and Z. Shi, “Towards automatic road mapping by fusing vehicle-borne multi-sensor data,” in *Proc. Int. Society for Photogrammetry and Remote Sensing Congress (ISPRS)*, 2008, pp. 867–872.
- [3] P. Doubek, M. Perdoch, J. Matas, and J. Sochman, “Mobile mapping of vertical traffic infrastructure,” in *Proc. Computer Vision Winter Workshop*, 2008.
- [4] M. Lehtomaki, A. Jaakkola, J. Hyypä, A. Kukko, and H. Kaartinen, “Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data,” *Remote Sensing*, vol. 2, pp. 641–664, 2010.
- [5] H. Yokoyama, H. Date, S. Kanai, and H. Takeda, “Detection and classification of pole-like objects from mobile laser scanning data of urban environments,” *Int. J. CAD/CAM*, vol. 13, no. 2, pp. 31–40, 2013.
- [6] S. El-Halawany and D. Lichti, “Detection of road poles from mobile terrestrial laser scanner point cloud,” in *Int. Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM)*, 2011.
- [7] A. Serna and B. Marcotegui, “Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning,” *ISPRS J. Photogramm. Remote Sens.*, apr 2014.
- [8] J. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert, “Scale selection for classification of point-sampled 3-d surfaces,” in *Proc. Int. Conf. on 3D Dig. Imag. and Modeling (3DIM)*, 2005, pp. 285–292.
- [9] F. Tombari and L. Di Stefano, “3d data segmentation by local classification and markov random fields,” in *Proc. Int. Conf. on 3D Imaging, Modeling, Visualization and Transmission (3DIMPVT)*, 2011.
- [10] M. Y. Yang and W. Förstner, “Plane detection in point cloud data,” Dep. of Photogrammetry, Institute of Geodesy and Geoinformation, University of Bonn, Tech. Rep. 1, Jan 25 2010. [Online]. Available: <http://www.ipb.uni-bonn.de/technicalreports/>
- [11] J. Kim and J. Pearl, “A computational model for combined causal and diagnostic reasoning in inference systems,” in *Proc. 8th Int. J. Conf. on Artificial Intelligence (IJCAI)*, 1983, pp. 190–193.
- [12] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice Hall, 2010.
- [13] A. Johnson, “Spin-images: A representation for 3-d surface matching,” Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, August 1997.
- [14] D. Muñoz, J. A. D. Bagnell, N. Vandapel, and M. Hebert, “Contextual classification with functional max-margin markov networks,” in *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2009.