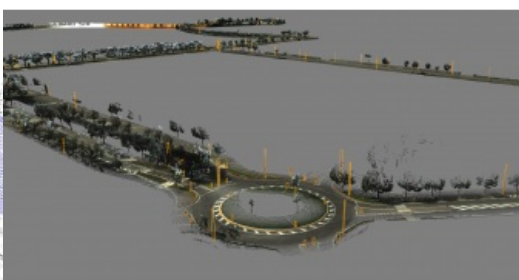
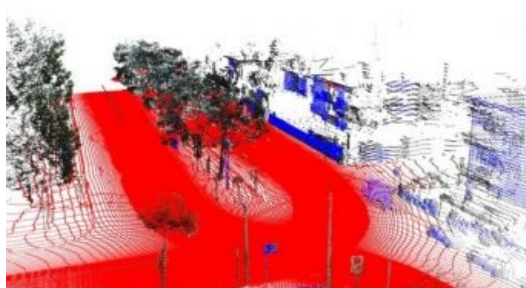


# AUTOMATIC EXTRACTION OF POLE-LIKE OBJECTS USING POINT CLOUD LIBRARY

## Poles from Point Clouds



Currently, the extraction of objects from point clouds of urban sites is commonly done manually as automation is impeded by noise, clutter, occlusions and varying point density. The authors have developed a software tool for automatic detection of road signs, lampposts, utility poles, traffic lights and other pole-like objects based on the open-source Point Cloud Library. The results appear to be

encouraging.

*By Federico Tombari, Tommaso Cavallari and Luigi Di Stefano, University of Bologna, Italy*

3D information about pole-like structures is used for detecting facility damage, road maintenance and other applications. Occlusion – the blocking of objects by other objects so that they are only partly visible in the point cloud – challenges the automatic detection of pole-like objects in point clouds. But even if a pole is represented by points from top to bottom, the number of points may be too small and the discrimination between poles and tree trunks or other thin-but-tall vertical objects may be difficult. Nevertheless, a method has been developed and implemented which is based on a two-stage classification – first local, then global (Figure 1).

### Data Reduction

If the point cloud has been acquired by laser scanning, the scans partly overlap. The result is that some parts of the point cloud contain twice as many points as other parts, i.e. the point density varies. In this method, to obtain uniform point density, point reduction is applied by allowing the presence of only one point per voxel (volume element: the 3D counterpart of pixel) of a pre-specified size. When setting the size of the voxel at  $1\text{cm}^3$ , the distance between two points in the reduced point cloud will be 1cm or more. There will also be many empty voxels which is inefficient when dealing with big datasets since storage requirements are proportional to the number of voxels, either filled or empty. An octree data structure avoids empty voxels needing storage capacity. This was created using the open-source Point Cloud Library (PCL) (see box).

### Removing Planes

To reduce the number of ambiguities and to speed up further processing, roads, building facades and other planar structures need to be detected and removed from the point cloud. This was done using a new customised RANSAC (Random Sample Consensus) algorithm, which iteratively fits a pre-specified model through points robustly, even in the presence of many outliers. The specification of the model uses the position of the points and their normal vectors, i.e. the vector perpendicular to the plane spanned up by the point itself and its nearest neighbouring points. The higher the number of neighbours, the smoother the estimation, although usually a number between 5 and 10 suffices. When the normal vectors of adjacent points are positioned in the same direction, these points most likely span up a plane. These points are removed to obtain a reduced point cloud used in the next steps. PCL is used to determine the normal vectors and a custom plane detection algorithm was written using PCL's Base class functionality.

### Local Classification

In a point cloud, a pole is represented as a cluster of points, which points toward the vertical direction. First, initial clusters, which may form part of larger clusters representing poles, are detected starting from individual points and searching for points in the vicinity. The search radius depends on the diameter of the pole and may be set to between 0.2cm and 1.5cm. In this method, the focus is on the angle formed by the upward direction and the line that connects the point with each neighbour; the idea is that the distribution of such angles around a pole is highly different to that around a structure which is not pole-shaped. In the case of a pole, most neighbouring points will be either above or below, and this is reflected in the angles. For each cluster, the angles are aggregated into a histogram. Although this local feature extraction method is not part of PCL, its Feature Class was used in writing a code, relying on the PCL's input/output facilities and its radius search on a KdTree. The subsequent steps also use functionality from PCL. Based on the histogram values, each point is classified as either pole or non-pole using Support Vector Machine (SVM), a supervised classification algorithm that can learn from training samples to

assign pre-defined classes to structures. The LIBSVM library is used as apparatus to carry out the classification. Training is done through providing labelled histograms of both poles and non-poles. Figure 3 shows examples of 4 pole and 4 non-pole histograms computed on real data, clearly showing the differences. Next, the histograms are put into SVM, which returns the likelihood that the point belongs to a pole. The high-probability point groups, which lie close to each other and together form a vertical structure, are clustered and the resulting larger clusters form the input for the global classification step.

## Global Classification

Clusters may be classified as belonging to a pole even though they are not – these are false positives. To remove such clusters, the centroid of a cluster is calculated and a plane is rotated around the principal direction of the cluster through the centroid. The size of the plane is set to the maximum distance between the centroid and the cluster points. While this rotating plane accumulates the coordinates of the intersecting points of the cluster, a second rotating plane – of the same size – will accumulate the coordinates of all points of the cloud (even those not belonging to the cluster). The additional information brought in by the latter plane provides context information to discriminate real poles from similar shapes such as tree trunks. These are then re-classified with SVM, resulting in the probability that the cluster represents an actual pole-like structure.

### PCL

*The method has been implemented in C++ using tools from Point Cloud Library (PCL), which is a large, open-source project for 2D/3D image and point cloud processing. It includes modules for filtering, feature estimation, recognition, registration, model fitting and segmentation. PCL emerged from the Robot Operating System (ROS) project launched by Willow Garage and is now being managed by the Open Perception Foundation, a non-profit spin-off from Willow Garage, backed up by many research institutions and companies from around the world.*

## Results

Tests were conducted on a georeferenced point cloud of 1.1 billion points acquired with a Topcon IP-S2 Compact + mobile mapping system (MMS) in the urban area of Verona, Italy, in 2012. Because of its size, the dataset was automatically subdivided into 1,294 blocks each representing a 15m trajectory. Each block was then subsampled into voxels of  $1\text{cm}^3$  resulting in an average of 860,000 points per block. Half of the data was used for training the SVM classifiers and cross-validation of the algorithm parameters, while the rest was used to evaluate the performance of the algorithm using manually labelled ground truth as reference. Overall, 75% of the poles were correctly identified with fewer than 20% false positives. Figure 4 shows an area with many poles correctly detected. Average overall processing time was 3.2 seconds per block. Processing time of local classification depends linearly on the number of points, and for global classification the dependency is quadratic. Hence, the subdivision into blocks helps to reduce processing time.

### Concluding Remarks

The method has the potential to be adapted for extracting a variety of pole-like structures and can be extended for automatic detection of other objects, such as building parts, vehicles and vegetation.

### Acknowledgments

Thanks are due to Samuele Salti and Alioscia Petrelli, and to QONSULT SpA for funding the research and granting the use of the 3D data.

### Further Reading

- Tombari, N. Fioraio, T. Cavallari, S. Salti, A. Petrelli, L. Di Stefano (2014) Automatic detection of pole-like structures in 3D urban environments, International Conference on Intelligent Robots and Systems.
- Tombari, L. Di Stefano (2011) 3D Data Segmentation by Local Classification and Markov Random Fields, International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT 2011)
- The Point Cloud Library (PCL) at [www.pointclouds.org](http://www.pointclouds.org)

*The original version of this article was published in the October 2014 issue of GIM International.*