Implementation and assessment of two density-based outlier detection methods over large spatial point clouds

Open Geospatial Data,
Software and Standards

ORIGINAL ARTICLE

Open Access

Implementation and assessment of two density-based outlier detection methods over large spatial point clouds

Francesco Pirotti[1,2*], Roberta Ravanelli[3], Francesca Fissore[1] and Andrea Masiero[1]

Abstract

Several technologies provide datasets consisting of a large number of spatial points, commonly referred to as point-clouds. These point datasets provide spatial information regarding the phenomenon that is to be investigated, adding value through knowledge of forms and spatial relationships. Accurate methods for automatic outlier detection is a key step. In this note we use a completely open-source workflow to assess two outlier detection methods, statistical outlier removal (SOR) filter and local outlier factor (LOF) filter. The latter was implemented ex-novo for this work using the Point Cloud Library (PCL) environment. Source code is available in a GitHub repository for inclusion in PCL builds.

Two very different spatial point datasets are used for accuracy assessment. One is obtained from dense image matching of a photogrammetric survey (SfM) and the other from floating car data (FCD) coming from a smart-city mobility framework providing a position every second of two public transportation bus tracks.

Outliers were simulated in the SfM dataset, and manually detected and selected in the FCD dataset. Simulation in SfM was carried out in order to create a controlled set with two classes of outliers: clustered points (up to 30 points per cluster) and isolated points, in both cases at random distances from the other points. Optimal number of nearest neighbours (KNN) and optimal thresholds of SOR and LOF values were defined using area under the curve (AUC) of the receiver operating characteristic (ROC) curve. Absolute differences from median values of LOF and SOR (defined as LOF2 and SOR2) were also tested as metrics for detecting outliers, and optimal thresholds defined through AUC of ROC curves.

Results show a strong dependency on the point distribution in the dataset and in the local density fluctuations. In SfM dataset the LOF2 and SOR2 methods performed best, with an optimal KNN value of 60; LOF2 approach gave a slightly better result if considering clustered outliers (true positive rate: LOF2 = 59.7% SOR2 = 53%). For FCD, SOR with low KNN values performed better for one of the two bus tracks, and LOF with high KNN values for the other; these differences are due to very different local point density. We conclude that choice of outlier detection algorithm very much depends on characteristic of the dataset's point distribution, no one-solution-fits-all. Conclusions provide some information of what characteristics of the datasets can help to choose the optimal method and KNN values.

* Correspondence: francesco.pirotti@unipd.it
[1]CIRGEO, Interdepartmental Research Center of Geomatics, University of Padua, Viale dell'Università 16, 35020 Legnaro, Italy
[2]TESAF Department, University of Padua, Viale dell'Università 16, 35020 Legnaro, Italy
Full list of author information is available at the end of the article

Introduction

Technologies related to acquisition of spatial data have grown exponentially and are still following this trend today. Spatial data are enabled when information recorded by the sensor is linked to a conventional spatial reference system, usually cartographically defined as a coordinate reference system (CRS). Such information is referred to as geoinformation. This allows to map the information from the CRS to the real world and viceversa. Global Navigation Satellite Systems (GNSS), before solely available for military applications from the United States' Global Positioning System (GPS) constellation, is now publicly accessible from several providers and with unprecedented accuracy. Accurate GNSS, along with a trend in the direction of lighter, less-expensive and metrically more accurate sensors, produces high-volumes of geospatial data. Crowd-sourcing solutions and sensors distributed in smart cities create and use large volumes of spatial data [1]. Datasets with unstructured points are a common direct or indirect output from such technologies.

Analyses of point-clouds has become a focus of scientific investigation also due to laser scanner technology. Laser scanners, from fixed, mobile or airborne platforms, can acquire several thousands of points per second, sampling objects and creating 3D representations. Technology in laser-derived 3D measurements is still improving at a fast rate; an example is the introduction of single photon-count sensors [2] which multiplies the number of measurements that a sensor can provide in a unit of time, potentially providing even larger datasets. Datasets with a large unstructured point can also be produced in a photogrammetric workflow, e.g. after aligning images using structure from motion (SfM), via dense matching [3] . The analysed datasets in this paper are derived from photogrammetry and from direct GNSS measurements, but the approach can be applied also to datasets from laser scanners.

In this scenario, outliers play an important role in the first phases of processing. A point dataset must be rid of outliers for the following modelling steps to be successful. Optimal outlier removal has been thoroughly investigated [4–8], and is still subject of investigation nowadays in many fields, such as fraud detection, medicine, pattern recognition and measurement error detection. Methods can be divided in supervised [9] and unsupervised: in this case the two tested methods belong to the unsupervised category.

Test data

Nowadays many spatially-enabled sensors can produce datasets with massive volume that can easily contain millions of points with attributes. In this study two quite different examples of such surveys were tested. One dataset is a product of a photogrammetric procedure (SfM) for creating a 3D model using overlapping imagery taken from a remotely piloted airborne system (RPAS). The second dataset is from trajectory data collected from vehicles every second via GNSS. These type of data are commonly referred to as Floating Car Data (FCD) and are becoming a very important part of smart-city frameworks.

SfM point dataset

A dense point dataset can be obtained from dense matching after reconstruction of a sparse 3D scene via photogrammetric SfM techniques [10]. This remote sensing method has strong support from open source libraries and software [11]. In this investigation, images where acquired with an RPAS flight carried out in July 2017 over an area with dense conifer forest, grass and some buildings as shown in Fig. 1. The final point density (ground sampling distance – GSD) is ~ 31 points per square meter. This dataset was chosen as it contains many characteristics that pose challenges to defining outliers: terrain surface has flat and steep parts, some areas have dense vegetation and others no vegetation, buildings or roads. From Fig. 1, top right, it is also evident that there are several outliers, i.e. points clearly not belonging to either the ground plane or the top surface. To have full control, outliers were manually removed to create a digital surface model (DSM) (Fig. 1 bottom left and right and Fig. 2 in green). Cloud Compare [12] was used to manually determine the clean DSM.

Artificial outliers were created to define a final control dataset (Fig. 2). Two types of outliers were created: (i) randomly positioned single points at a distance between 1 and 200 m from the DSM and (ii) randomly positioned clusters of points, with 2 to 30 points per cluster, with the cluster centre randomly positioned between 2 and 200 m above the DSM (Fig. 2 in red and blue respectively). R cran [13] was used to simulate and add the outliers to the dataset by randomly picking a non-outlier point and transforming its position according to the rules described above.

FCD – Floating Car data

The largest part of movements in an urban environment is constrained to the road network. Thanks to the recent development of navigation technologies, nowadays GNSS sensors represent a low-cost, efficient and already largely widespread tool to collect such movement information from different types of objects, including pedestrians and vehicles (cars, bicycles, buses …) [14], especially if compared with more traditional traffic monitoring methods like loop detectors or automatic plate number recognition [15]. GNSS sensors are capable of recording at high rate, e.g. 1 position per second of the tracked object, so that its

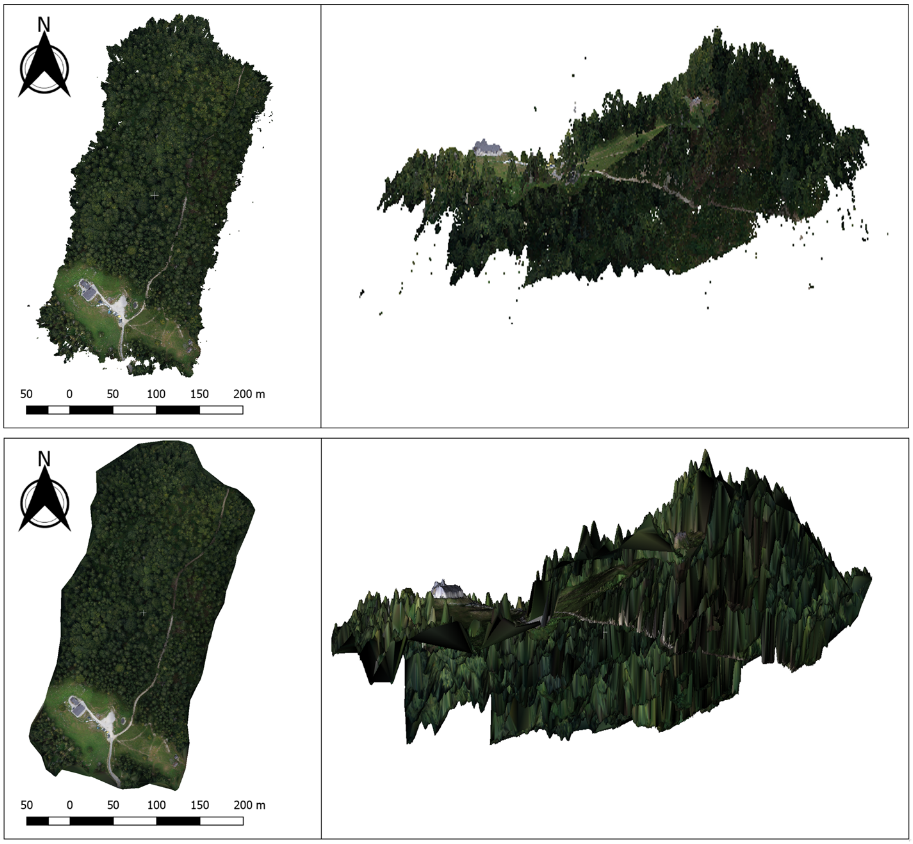


Fig. 1 Point clouds from RPAS flight – outliers are clearly visible on the top, cleaned dataset is shown in the bottom as meshed surface

continuous movement is recorded as a trajectory containing a sequence of sampled points. This type of surveying is extremely important in estimating hazard situations, e.g. integrated with remote sensing [16] or integrated with geographic information systems (GIS) [17, 18].

These type of data are gaining importance as new paradigms are being implemented in real scenarios. Bigdata processing for smart-cities can be applied to high volumes of data from multiple sensors, which are analysed to get in depth information on the multiple dynamic aspects of a mobility and other factors.

Such data can be corrupted by noise [15] due to the pretty well-known problems encountered by GNSS in urban environment (e.g. obstructions, multipath). Critical information can be extracted if a proper preliminary data cleaning for possible spurious data/outliers is performed. To underline this key step, in this work the FCD of the city of Turin (Italy) Public Transportation system were analysed. The preliminary step for the impedance map calculation consisted in the removal of all the information not referable to the actual path of the lines- see Fig. 3.

To test outlier detection the FCD from two bus lines were used, line 11 and line 39. The methods were applied to 2D and 3D data: 2D dimensions were geospatial positions, i.e. latitude and longitude provided by GNSS,

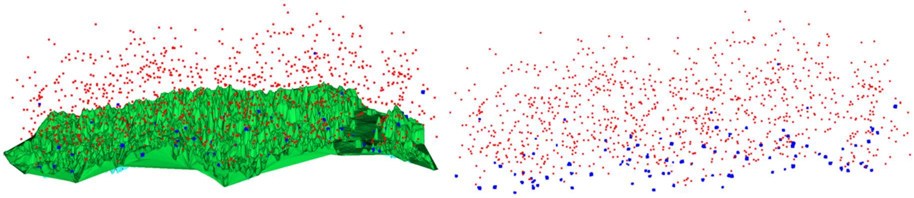


Fig. 2 SfM dataset – left: clean DSM surface in green, random unclustered outliers in red, clustered random outliers in blue – right: only outliers

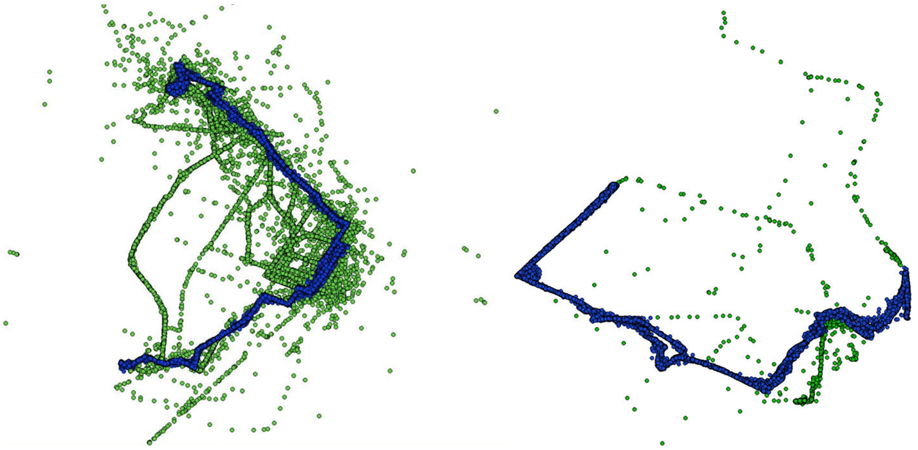


Fig. 3 FCD of BUS lines in urban environment. Right is BUS line 11, left is BUS line 39 – outliers are in green

and the third dimension was the estimated velocity of the vehicle at each point.

Methods

There are many outlier detection methods in literature, in this study case we focus on unsupervised methods based on local density metrics of points. The rationale behind the two tested methods is that in large datasets consisting of 3D points the number of outliers is much lower than the number of correct points. The correct points are also clustered with respect to outliers, and therefore outliers can be detected by metrics that represent mutual distance between neighbouring points. In the next sub-sections the two methods are described in-depth.

From the definition by Hawkings [5] "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". In point datasets from laser scanners, SfM and other spatial sensors outliers can be produced from incorrect processing, multipath or from unwanted objects [4], such as birds or dust particles. In SfM in particular, which represents the first dataset, outliers can be from mismatches of keypoint descriptors, which can be common when using a small number of targets or none at all – e.g. with smartphones, or where the image geometry is below optimal [19, 20].

There are several ways to remove outliers with unsupervised, semi-unsupervised or even manual methods. Many users still prefer to remove outliers manually [21], but in this implementation the target is to have a high degree of automation, therefore the two methods that were tested are unsupervised.

In this implementation we tested two methods: (i) Statistical Outlier Removal (SOR), (ii) local outlier factor (LOF). Four predictors – two per each method – were tested: they consist of SOR and LOF values for each point, and of absolute differences, with respect to their median value, of LOF and SOR values, referred to as SOR2 and LOF2 respectively. The hypothesis behind these last two predictors is that most points will be correct, and the median of the distribution of SOR and LOF values will reflect correctness, thus points with values of SOR or LOF distant from the median will likely be outliers. The threshold for optimal results is calculated using ROC curves and applied to flag outliers.

All the above described methods require detecting a number K of nearest neighbours (KNN) to each point. The creation of a metric structure in large point sets is critical for detection of KNN in an acceptable time span. K-d tree structures and methods for approximate nearest neighbours search are implicitly used in the implementation of the methods, *libnabo* for R cran [22] and the fast library for approximate nearest neighbours (FLANN) [23] in the point cloud library (PCL) [24].

Statistical outlier removal (SOR)

The SOR method is a distance-based approach, which assigns a probability of being an outlier to each point by

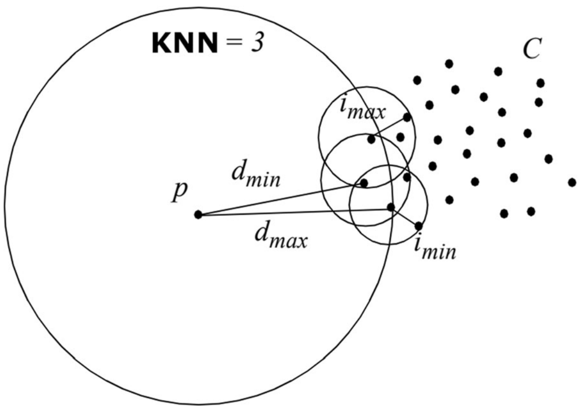


Fig. 4 Schema of nearest neighbours with LOF reachability distance for KNN = 3 (modified from Breunig et al. [18])

comparing its distance to neighbours. The statistic used in this case is local density calculated by measuring distances of a user-defined number K of nearest neighbours [8] (in this paper referred to as KNN). It is trivial to state that outliers, by definition, should be significantly distant from the main distribution of other points, see Fig. 4. SOR filter for this work was implemented as an R function using *nabor* package [22] for fast calculation of KNN distances. SOR filter is also fully implemented as part of PCL.

Local outlier factor (LOF)

The local outlier factor (LOF) algorithm as described by [25, 26] is an unsupervised method which assigns a score to each point by computing its local density deviation with respect to its neighbours in a cluster. An outlier or a group of outliers substantially have a lower density than their neighbours do, thus a LOF value significantly greater than the rest (see Eq. 1–4).

The number of neighbours chosen is typically greater than the minimum number of points a cluster can contain, so that other points can be local outliers relative to this cluster. In practice, such information can be available if the user is knowledgeable about the data. Such situation is likely in the two presented cases, as SfM point density and GNSS rate of recording can provide estimation of respective point density. The LOF method also has the advantage of limiting statistical fluctuations [27].

Fundamentally three steps are necessary to extract LOF values for each point. First for each point (i) every distance with *k* other points is calculated, and defined as K.dist.

$$K.dist_{i,j} = dist\left(P_i, P_j\right) \tag{1}$$

where K-distance of point P_i is the distance between P_i and K^{th} nearest point, P_j .

The second step calculates reachability distance (R.dist) for every point and its K neighbours. The reachability distance is the maximum between two values: the K.dist of the considered point and the considered neighbour, for each KNN other points (see Fig. 4).

$$R.dist(P_i, P_{K^{th}}) = \max(K.dist_{K^{th}}(P_{K^{th}}); K.dist_i) \tag{2}$$

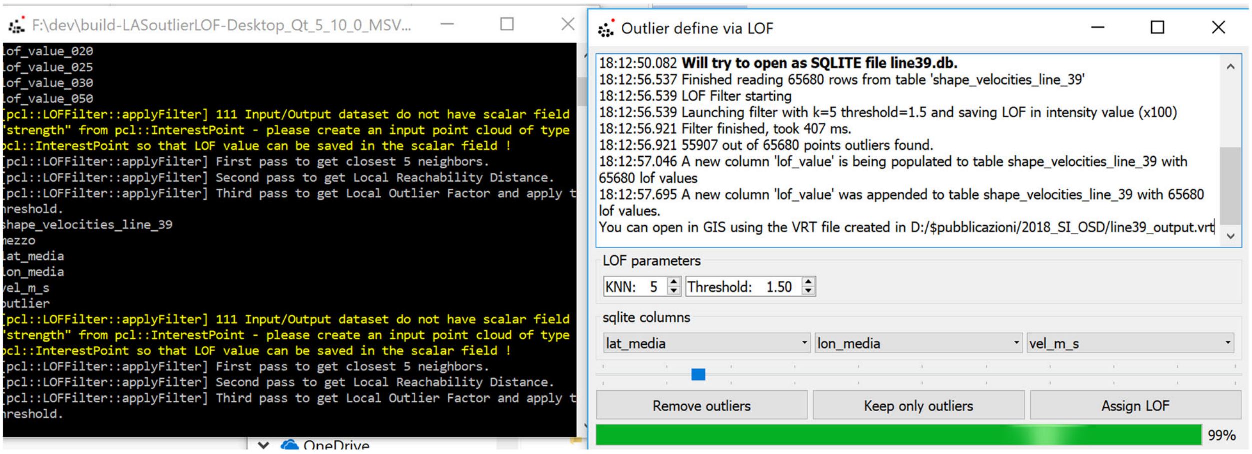
The local reachability density (LRD) is then defined for each point as inverse of the average reachability distances of point P_i. In the equation below, the numerator defines the cardinality of the point set of KNN.

$$LRD(P_i) = \frac{\|N_k(P_i)\|}{\sum_{P_j \in N_k(P_i)} R.dist\left(P_i, P_j\right)} \tag{3}$$

The last step calculates LOF value for each point is calculated by comparing LRD value of the point with LRD value of its *k* neighbours.

$$LOF(P_i) = \frac{\sum_{P_j \in N_k(P_i)} \frac{LRD\left(P_j\right)}{LRD(P_i)}}{\|N_k(P_i)\|} \tag{4}$$

In this work the LOF method is implemented as a new filter in point cloud library (PCL). The source code is available in a GitHub repository for inclusion in PCL builds [28]. PCL is a “standalone, large scale, open project for 2D/3D image and point cloud processing. PCL is released under the terms of the BSD license, and thus free for commercial and research use” [24, 29]. PCL provides the ideal framework to process large point datasets. In these methods finding nearest neighbours is an essential step. Spatial metric structures allow approximate nearest neighbours matching with binary trees and are implemented in PCL via the FLANN library [30, 31].



```
lof_value_020
lof_value_025
lof_value_030
lof_value_050
[pcl::LOFFilter::applyFilter] 111 Input/Output dataset do not have scalar field
"strength" from pcl::InterestPoint - please create an input point cloud of type
pcl::InterestPoint so that LOF value can be saved in the scalar field !
[pcl::LOFFilter::applyFilter] First pass to get closest 5 neighbors.
[pcl::LOFFilter::applyFilter] Second pass to get Local Reachability Distance.
[pcl::LOFFilter::applyFilter] Third pass to get Local Outlier Factor and apply t
hreshold.
shape_velocities_line_39
mezzo
lat_media
lon_media
vel_m_s
outlier
[pcl::LOFFilter::applyFilter] 111 Input/Output dataset do not have scalar field
"strength" from pcl::InterestPoint - please create an input point cloud of type
pcl::InterestPoint so that LOF value can be saved in the scalar field !
[pcl::LOFFilter::applyFilter] First pass to get closest 5 neighbors.
[pcl::LOFFilter::applyFilter] Second pass to get Local Reachability Distance.
[pcl::LOFFilter::applyFilter] Third pass to get Local Outlier Factor and apply t
hreshold.
```

```
18:12:50.082 Will try to open as SQLITE file line39.db.
18:12:56.537 Finished reading 65680 rows from table 'shape_velocities_line_39'
18:12:56.539 LOF Filter starting
18:12:56.539 Launching filter with k=5 threshold=1.5 and saving LOF in intensity value (x100)
18:12:56.921 Filter finished, took 407 ms.
18:12:56.921 55907 out of 65680 points outliers found.
18:12:57.046 A new column 'lof_value' is being populated to table shape_velocities_line_39 with
65680 lof values
18:12:57.695 A new column 'lof_value' was appended to table shape_velocities_line_39 with 65680
lof values.
You can open in GIS using the VRT file created in D:/$pubblicazioni/2018_SI_OSD/line39_output.vrt
```

LOF parameters

KNN: 5 Threshold: 1.50

sqlite columns

lat_media lon_media vel_m_s

Fig. 5 LOF implementation GUI

Software implementation

A PCL build was integrated in a graphical user interface (GUI) for processing two data formats: LAS/LAZ point clouds and SQLite database format. The former was used for the SfM dataset and implemented using LASlib (with LASzip), a "C++ programming API for reading / writing LIDAR data stored in standard LAS or in compressed LAZ format (1.0 - 1.3)" [32]. Both LASlib and LASzip are released under the terms of the GNU Lesser General Public Licence. The latter format, SQLite, was used to read FCD data, and was implemented using the dedicated library in public domain: "SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine" [33]. The GUI was developed in C++ using the Qt Framework IDE. The PCL build included the local implementation of the LOF method and thus it was applied to the analysed point datasets via the GUI. The GUI also provides information on the process via a log (see Fig. 5).

Results and discussion

It is trivial that the best method and combination of parameters (KNN and threshold) must have the highest number of true positives and true negatives and the lowest number of false positives and false negatives. In this investigation we consider detecting points which are outliers, therefore positives are the outliers and negatives

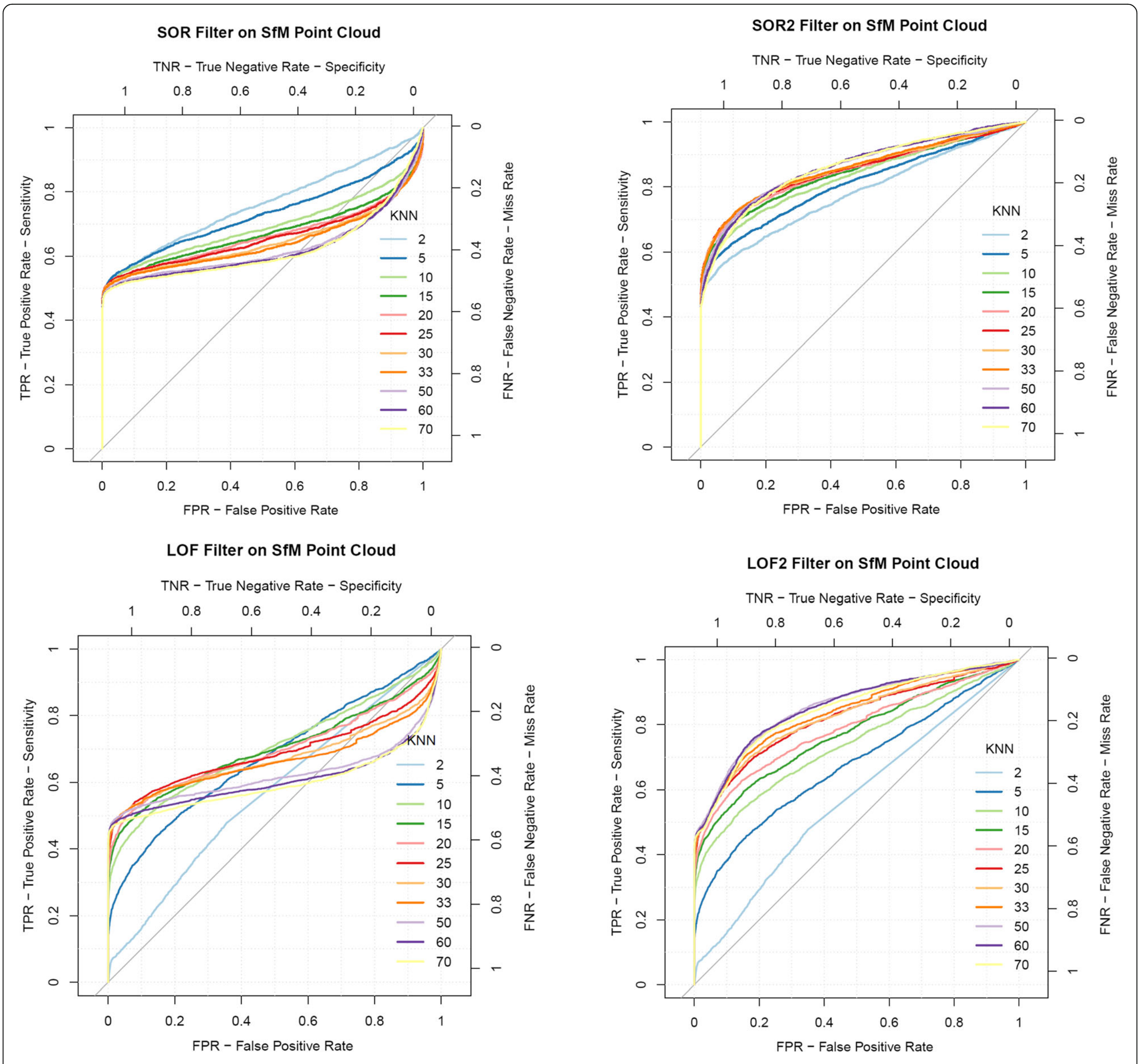


Fig. 6 ROC Curves of the four predictors

are the inliers. Two possible types of errors can be present when predicting a binary response (inliers vs. outliers): false outliers (i.e. type I error, false positive rate - FPR), and false inliers (i.e. type II error, missed outliers, false negative rate- FNR). In this investigation particular attention is given to false inliers (FN) – points which are outliers, but are incorrectly assigned as inliers, are considered. This is because for further processing of point datasets, this type of error leads to worse consequences than false outliers. The Receiver Operating Characteristic (ROC) curve is used to define optimal balance overall performance and best-performing threshold, and false negative rate is analysed in depth.

ROC curves

"The Receiver Operating Characteristic (ROC) curve is used to assess the accuracy of a continuous measurement for predicting a binary outcome" [34]. In particular, it allows to intuitively evaluate metrics at varying thresholds; this is exactly what is looked for in this study case, where continuous metrics are used to discriminate between outliers and inliers (i.e. SOR and LOF). ROC curves have long been used in signal detection theory and applications [35]. As mentioned, for a predictor consisting in single continuous measurements, convention dictates that a test positive for outlier is defined as the value of the predictor (LOF or SOR) of a point exceeding a fixed threshold (T):

$$\begin{aligned} T &= threshold = \phi \in \mathbb{R} \\ &\phi \in [V_{\min}, V_{\max}] \end{aligned} \tag{5}$$

where V is the value of LOF, LOF2, SOR or SOR2: V_{min} is the lowest and V_{max} is the highest value in the set.

The two axes of the ROC graph are respectively:

$$\begin{aligned} ROC_{x1}(\phi) &= \mathrm{FPR}_{outlier}(\phi) = \frac{FP(\phi)}{FP(\phi) + TN(\phi)} \\ ROC_{x2}(\phi) &= \mathrm{TNR}_{outlier}(\phi) = 1\text{–}\mathrm{FPR}_{outlier}(\phi) \\ ROC_{y1}(\phi) &= \mathrm{TPR}_{outlier}(\phi) = \frac{TP(\phi)}{FN(\phi) + TP(\phi)} \\ ROC_{y2}(\phi) &= \mathrm{FNR}_{outlier}(\phi) = 1\text{–}\mathrm{TPR}_{outlier}(\phi) \end{aligned} \tag{6}$$

Since the threshold T has to be determined, we plot TPR as a function of FPR for all possible values V. This will be applied to the SfM dataset and to the two FCD datasets to determine the optimal value of T for all cases. Optimal T is chosen by adopting the corresponding value of T which provides the highest value of area under the curve (AUC). The AUC is a single combined measure of sensitivity and specificity allowing effective comparison between results [36]. Specific results for the two datasets are reported in the next sections.

SfM point dataset

The plots in Figs. 6 and 7 allow interpretation and discussion of the performance of the four predictors applied to the SfM dataset. The overall best performance was by SOR2 and LOF2. In both cases higher values of KNN improve accuracy, up to KNN = 50 where there is only very slight improvements.

Figures 6 and 7 define SOR2 and LOF2 as having similar accuracies when considering detection of outliers, but, as mentioned in data description, in the SfM dataset two classes of outliers were artificially added, clustered and unclustered (see Fig. 2). A more in depth analysis can thus be carried out to assess if methods behave differently with respect to the two classes of outliers. The adopted method of manually inserting outliers allows us to have full control; e.g. we know the distance from the DSM, thus we could test if there is correlation between

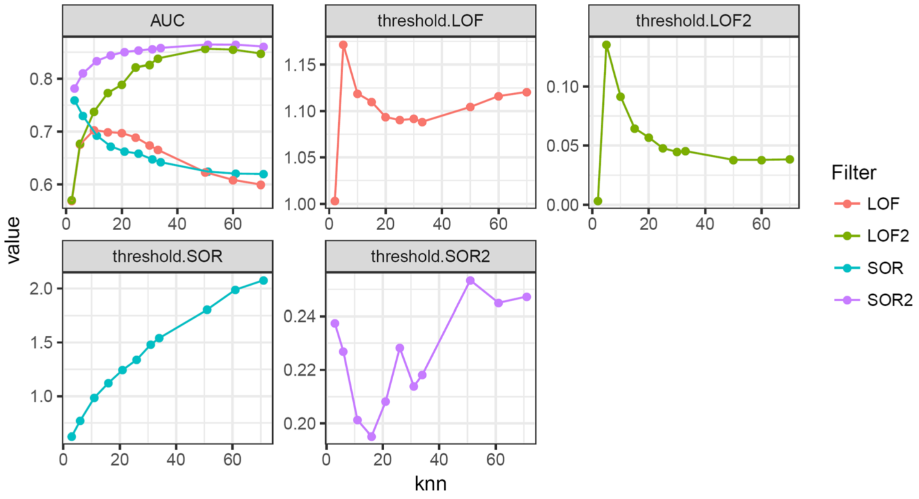


Fig. 7 AUC values and best threshold values calculated from ROC curves at different KNN values for all four predictors

Table 1 Number of undetected (FN) outliers and false negative rate in the two classes, clustered and unclustered (see Fig. 2)

	FN (n. of points)		FNR	
	Clustered	Unclustered	Clustered	Unclustered
CONTROL	1596	1000		
SOR	1208	0	0.760	0.000
SOR2	750	1	0.470	0.001
LOF	994	324	0.623	0.324
LOF2	643	3	0.403	0.003

detection rate and distance from the inliers in the point dataset. Table 1 reports the number of undetected (FN) outliers for each class and Table 2 provides the more complete confusion matrix. It is worth noting that the two methods show different behaviour with respect to the outlier class. Almost all of the unclustered outliers were detected by SOR, SOR2 and LOF2, whereas LOF had very low detection rate. The opposite is true when considering clustered points, LOF2 provides the best result (lower FNR – see Table 1). The reachability distance used in LOF to determine local density is a measure to produce more stable results within clusters [25]. This is clearly inferred by the low AUC values which result when KNN is too low (Fig. 9); i.e. when KNN is below the number of points per cluster that was set in the artificially clustered outliers (30 points), AUC is low. On the other hand, accuracy increases significantly in LOF2 when KNN surpasses this value.

Threshold values allow getting insights on what point density determines what an inlier is and what an outlier is. SOR2, is the overall best method, with, at best KNN value (Fig. 7 - KNN = 60), the threshold being ~ 0.25 – meters is the unit in this case. It is to be interpreted as the average of distances, between the point and 60 nearest neighbours; this value determines if the point is an inlier (below 0.25) or an outlier (above 0.25). LOF values represent local density with values near 1 being considered inliers, and values above tend to indicate outliers [25]. This is also intuitively seen in Fig. 8 where the median values of LOF values of all points (1 million inliers and 2596 outliers) are very close to 1. In our case, regarding the SfM dataset, the LOF value threshold is 1.1 at its best KNN value (Fig. 7 - KNN = 10). This value reflects results from literature, where, for datasets with low local fluctuations, LOF values above 1.1 are likely to be outliers [25], whereas, in other types of datasets with varying densities, i.e. high local fluctuations, higher LOF values might still indicate inliers. SOR2 and LOF2 thresholds are defined by distance from the respective medians, which are shown in Fig. 8.

LOF2 performed close to SOR2 and both outperformed LOF and SOR. This indicates that assigning to each point a metric based on absolute difference from median, improves the ability to discern outliers from inliers.

FCD – Floating Car data

Figure 9 summarizes results from ROC curves of FCD data by providing AUC values at different KNN values. It is clear that the different point distribution (Fig. 3) impacts on which method and which KNN provides the highest outlier detection rate (TPR – see Table 3). Also

Table 2 Confusion matrix of results where two outlier classes, clustered and unclustered (see Fig. 2) are defined in the control and matched against results from the four predictors – in green the correct number of points detected outliers/inliers (true positives and negatives respectively): percentages represent the true positive rate

		CONTROL		
		inliers	*outliers*	
			Clustered	Unclustered
SOR	*inliers*	958524	1208	0
	outliers	41066	388(24.0%)	1000(100%)
SOR2	*inliers*	891653	750	1
	outliers	107937	846(53.0%)	999(99.9%)
LOF	*inliers*	893071	994	324
	outliers	106519	602(37.7%)	676(67.6%)
LOF2	*inliers*	813340	643	3
	outliers	186250	953(59.7%)	997(99.7%)

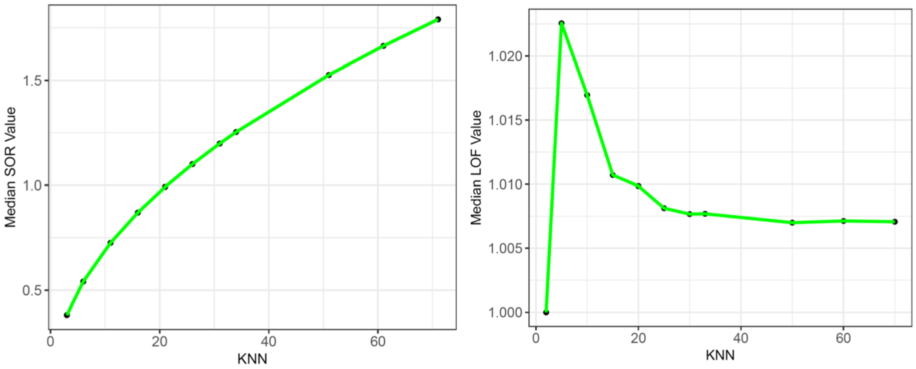


Fig. 8 Median values in SOR and LOF distributions at different values of KNN

the overall performance differed for each line. Line 11 had the best results with low KNN (Fig. 9 - KNN = 3) using SOR or SOR2 methods; using the third dimension did not significantly change the results of AUC values or the values of TPR. Line 39 had lower AUC values and TPR values, but best method resulted to be LOF, with highest KNN of 70 neighbours, leaving out the third dimension, i.e. the velocity of the vehicle. It is worth noting that velocity either did not contribute to improve accuracy, in line 39 it even decreased accuracy, so this metric is not useful in case we want to predict points that do not belong to the original route.

Figure 10 is a visual representation of Bus line 11 before and after the application of the SOR2 method with KNN = 3. It is not clear in the image as it is 2D, but several overlapping points are present in what looks like an isolated point outside the main track.

Conclusions

Two objectives were reached in the presented investigation: the implementation of the LOF method in the PCL open-source library with its integration in a GUI, and results of testing the LOF method against the SOR method using two very diverse datasets in terms of technology and point density and distribution. It is worth noting that investigations on outlier detection methods keeps on being a topic of high interest, due to the many technologies that provide datasets with a large number of unstructured points.

Results are mixed, with the two datasets resulting in best performances from different methods and threshold types. This indicates that, very likely, the type of point distribution, i.e. the local density fluctuation, influences on the choice of method for detecting outliers. SfM point dataset clearly LOF2 performed close to SOR2, both with high KNN values, and both outperformed LOF and SOR. This indicates that assigning to each point a metric based on absolute difference from median, improves the ability to discern outliers from inliers. This is quite different from the FCD datasets; which showed opposite behaviour. The best results were given by low values of KNN for all except the 2D dataset of line 39, which had highest KNN perform best. SOR performed best for line 11 whereas line 39 had SOR2 at lowest KNN do best for the 3D dataset, and LOF do best for the 2D dataset; again with lowest and highest KNN

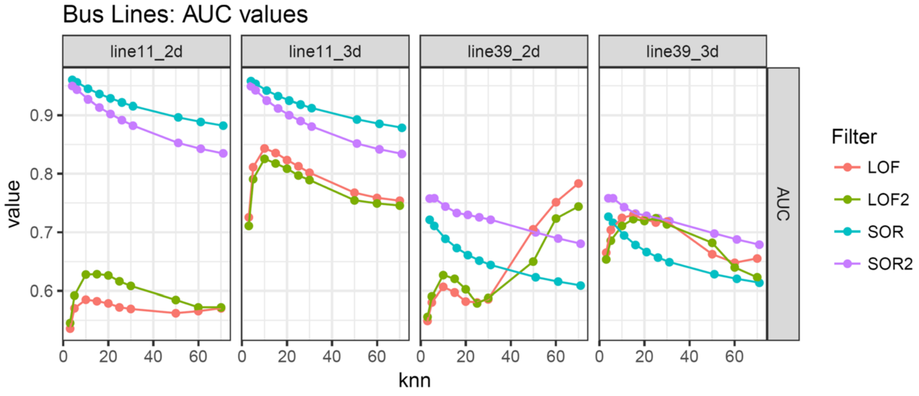


Fig. 9 AUC values of two bus lines, using only latitude and longitude (2d) and also velocity (3d); LOF2 and SOR2 use normalized values as predictors

Table 3 True positive rate – TPR – for different bus lines and methods at optimal KNN (see Fig. 9); in green the best relative to each line and 2d/3d combination, with thick border the overall best for each bus line

	SOR	SOR2	LOF	LOF2
Line11_2d	91.5%	92.8%	38.3%	52.3%
Line11_3d	92.0%	92.7%	70.1%	69.2%
Line39_2d	54.0%	54.1%	68.1%	67.6%
Line39_3d	53.0%	63.1%	59.8%	60.0%

respectively. This seemingly erratic behaviour reflects the very different datasets chosen for testing, which was one of the objectives of this investigation. As mentioned, SfM has a much more consistent density, whereas FCD has higher density fluctuations. This can explain why thresholds of absolute differences from the median (SOR2 and LOF2) outperformed with respect to using LOF and SOR values as thresholds, whereas this was not the case for the FCD dataset. It is worth mentioning that points at border of a dataset can be perceived as outliers, but this case can be considered a "margin" effect that can be ignored in most cases because the objects of interest in a survey are usually not at the margin of the survey; this is to be considered when planning a survey.

An aspect worth noting is that in SfM dataset the AUC value for best methods (LOF2 and SOR2) levels out at higher KNN values. This is important because it indicates that result at the best KNN = 60 is no particularly better than the result from KNN = 20. Considering that processing is much faster at the latter value of KNN, users can choose this value instead of the higher value. Another interesting point is that at and above KNN = 20 results are good, and they seem to stabilize, i.e. results do not deteriorate with higher KNN values. Experimentation stopped at KNN = 70, also due to long processing time, future tests might increase KNN to see if, and when, there is a deterioration. This behaviour is likely related to the median value of LOF (Fig. 8 - right) that becomes stable at KNN > = 20, meaning that at least 20 neighbours are necessary, for the SfM dataset, to represent the local fluctuation. In SfM dataset, while LOF2 increases with KNN, LOF is constant at KNN 10–20 and deteriorates at KNN > 20. In this dataset KNN values in the 10–20 range bring this difference between LOF and LOF2, likely due to the way that different thresholds are calculated; i.e. using, as threshold, the absolute difference from median LOF improves the efficiency of the method, whereas LOF value alone is not enough to discriminate outliers from inliers.

Other practical considerations are necessary to select the proper approach for removing outliers. The dataset must be analysed to understand if there are any systematic ways to model either outliers or inliers. For example SfM datasets are more prone to have outliers related to the Z axis value, whereas the floating car dataset has outliers which are sensible to planar offsets due to vehicles going on different routes with respect to the usual track. Therefore a careful evaluation of the dataset source will help to figure which descriptors can be inserted to improve results. In the FCD point dataset, the third dimension is velocity, but this feature did not improve results with respect to only planar 2D spatial coordinates. It is very likely that better results can be

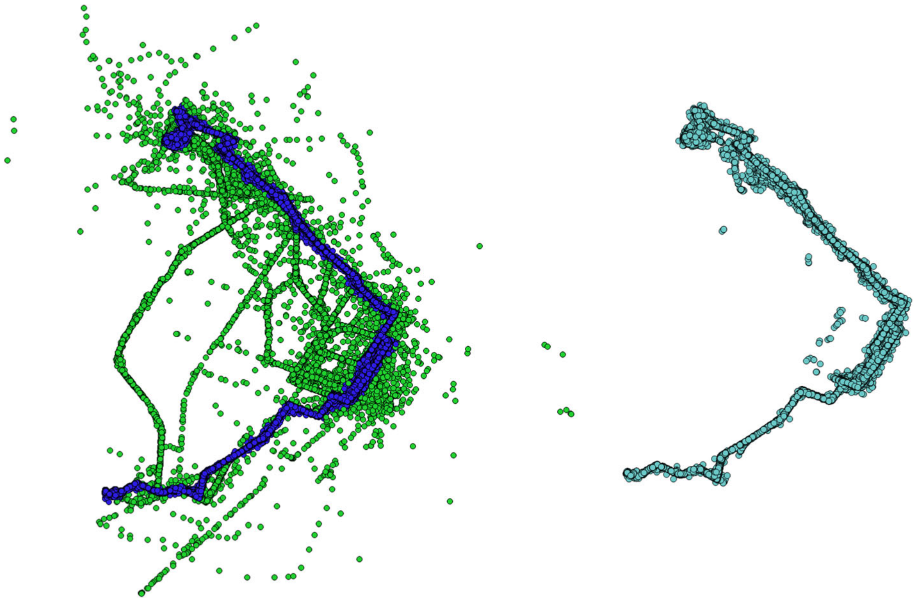


Fig. 10 Bus line 11 before and after applying the SOR filter using two dimensions (latitude and longitude)

achieved with specific descriptors extracted from the dataset. For example, the floating car dataset has a linear characteristic; therefore, a degree of linearity of neighbouring points can be added as descriptor and will likely improve results. The focus of this paper is to assess two generic algorithms and not to evaluate specific use cases, but it is worth reporting that specific descriptors can help in detecting outliers.

The bottom-line of the results is that there is not a one-method-suits-all, and not a best number of nearest neighbours - KNN - to consider in these two methods. Best KNN values strongly depend on local density of points. As mentioned, to choose ideal KNN, enough neighbours must be used to represent local fluctuations. This seems trivial, but is important to keep in mind. Differences in AUC and TPR values show that ideal combinations of method and KNN must be chosen depending on the characteristics of the dataset and of the type of outliers that are expected (clustered or not).

Acknowledgements
The data on public transportation (BUS trajectories) were provided by Prof. Piero Boccardo, Polytechnic of Torino, part of the URBAN-GEO BIG DATA project which funded this investigation (see section on funding).

Funding
This work is funded and supported by URBAN-GEO BIG DATA, a Project of National Interest (PRIN) funded by the Italian Ministry of Education, University and Research (MIUR) – id. 20159CNLW8.

Availability of data and materials
Data are available upon request to the corresponding author. Original source code of LOF implementation in PCL library is available as open source (GNU GPL) in GitHub [28].

Authors' contributions
FP created the main idea and the structure of methods, RR organized and implemented the method in the FCD dataset, FF contributed to discussion and review, AM verified algorithms and supported review. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]CIRGEO, Interdepartmental Research Center of Geomatics, University of Padua, Viale dell'Università 16, 35020 Legnaro, Italy. [2]TESAF Department, University of Padua, Viale dell'Università 16, 35020 Legnaro, Italy. [3]Geodesy and Geomatics Division, DICEA - University of Rome "La Sapienza", Rome, Italy.

Received: 30 April 2018 Accepted: 22 August 2018
Published online: 10 September 2018

References

1. Brovelli MA, Minghini M, Zamboni G. New generation platforms for exploration of crowdsourced geo-data. In: Earth Observation Open Science Innovation. Cham: Springer International Publishing; 2018. p. 219–43. Available from: https://doi.org/10.1007/978-3-319-65633-5_9.
2. Swatantran A, Tang H, Barrett T, DeCola P, Dubayah R. Rapid, High-Resolution Forest Structure and Terrain Mapping over Large Areas using Single Photon Lidar. Sci Rep. 2016;6:28277. Available from: http://www.nature.com/articles/srep28277
3. Remondino F, Barazzetti L, Nex F, Scaioni M, Sarazzi D. UAV photgrammetry for mapping and 3D modeling – current status and future perspectives. Int. Arch Photogramm Remote Sens Spat Inf Sci. 2011;38:14–6.
4. Sotoodeh S. Outlier Detection in Laser Scanner Point Clouds. Int Arch Photogramm Remote Sens Spat Inf Sci. 2006;36:297–302. Available from: http://www.isprs.org/proceedings/XXXVI/part5/paper/SOTO_653.pdf
5. Hawkins DM. Identification of Outliers. Dordrecht: Springer Netherlands; 1980. https://doi.org/10.1007/978-94-015-3994-4.
6. Hodge VJ, Austin J. A survey of outlier detection Methodologies. Artif Intell Rev. 2004;22:85–126. Available from: http://link.springer.com/article/10.1007/s10462-004-4304-y
7. Atanassov R, Bose P, Couture M, Maheshwari A, Morin P, Paquette M, et al. Algorithms for optimal outlier removal. J Discret Algorithms. 2009;7:239–48. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1570866709000021
8. Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. ACM Sigmod Rec. 2000;29:427–38. Available from: http://dl.acm.org/citation.cfm?id=335437
9. Pirotti F, Sunar F, Piragnolo M. Benchmark Of Machine Learning Methods for Classification of a Sentinel-2 Image. Int Arch Photogramm Remote Sens Spat Inf Sci. 2016;41:335–40. Available from: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B7/335/2016/
10. Masiero A, Fissore F, Pirotti F, Guarnieri A, Vettore A. Toward the use of smartphones for mobile mapping. Geo-spatial Inf Sci. 2016;19:210–21.
11. Pirotti F, Neteler M, Rocchini D. Preface to the special issue "Open Science for earth remote sensing: latest developments in software and data.". Open Geospatial Data Softw Stand. 2017;2:26. Available from: http://opengeospatialdata.springeropen.com/articles/10.1186/s40965-017-0039-y
12. Girardeau-Montaut D. CloudCompare (version 2.9) [GPL software] [Internet]. 2017. Available from: http://www.cloudcompare.org/. Accessed 01 Jan 2018.
13. Bivand RS, Pebesma E, Gomez-Rubio V. Applied spatial data analysis with R. 2nd ed. New York: Springer; 2013.
14. Guarnieri A, Pirotti F, Vettore A. Low-cost MEMS sensors and vision system for motion and position estimation of a scooter. Sensors. 2013;13:1510–22. Available from: http://www.mdpi.com/1424-8220/13/2/1510/
15. Yang C, Gidófalvi G. Mining and visual exploration of closed contiguous sequential patterns in trajectories. Int J Geogr Inf Sci. 2018;32(7):1282–304.
16. Boccardo P, Tonolo FG. Remote sensing role in emergency mapping for disaster response. In: Eng. Geol. Soc. Territ. - Vol. 5 Urban Geol. Sustain. Plan. Landsc. Exploit; 2015.
17. Pirotti F, Brovelli MA, Prestifilippo G, Zamboni G, Kilsedar CE, Piragnolo M, et al. An open source virtual globe rendering engine for 3D applications: NASA World Wind. Open Geospatial Data Softw Stand. 2017;2:4. Available from: http://opengeospatialdata.springeropen.com/articles/10.1186/s40965-017-0016-5
18. Piragnolo M, Pirotti F, Guarnieri A, Vettore A, Salogni G. Geo-Spatial Support for Assessment of Anthropic Impact on Biodiversity. ISPRS Int J Geo-Information. 2014;3:599–618. cited 2014 Apr 26]. Available from: http://www.mdpi.com/2220-9964/3/2/599
19. Barazzetti L, Remondino F, Scaioni M. Automation in 3D reconstructing results on different kinds of close-range blocks. Int Arch Photogramm Remote Sens Spat Inf Sci. 2010;38:55–61.
20. Scaioni M, Feng T, Barazzetti L, Previtali M, Lu P, Qiao G, et al. Some applications of 2-D and 3-D photogrammetry during laboratory experiments for hydrogeological risk assessment. Geomatics Nat Hazards Risk. 2014 [cited 2014 Jun 28:1–24. Available from: http://www.tandfonline.com/doi/abs/10.1080/19475705.2014.885090
21. Westoby MJ, Brasington J, Glasser NF, Hambrey MJ, Reynolds JM. 'Structure-from-motion' photogrammetry: a low-cost, effective tool for geoscience applications. Geomorphology. 2012;179:300–14. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0169555X12004217
22. Elseberg J, Magnenat S, Siegwart R, Nüchter A. Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. J Softw Eng Robot. 2012;3:2–12.
23. Muja M, Lowe DG. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. Int Conf Comput Vis Theory Appl Viss. 2009:331–40.
24. PCL Point Cloud Library. 2017. Available from: http://pointclouds.org/. Accessed 01 Jan 2018.
25. Breunig MM, Kriegel H-P, Ng RT, Sander J. LOF: Identifying Density-Based Local Outliers. In: Proc. 2000 Acm Sigmod Int. Conf. Manag. Data; 2000. p.

1–12. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.8948.

26. Kriegel H-P, Kröger P, Schubert E, Zimek A. LoOP: local outlier probabilities. In: Proc. 18th ACM Conf. Inf. Knowl. Manag; 2009. p. 1649–52. Available from: http://doi.acm.org/10.1145/1645953.1646195.
27. Goldstein M, Uchida S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS One. 2016;11:e0152173.
28. Pirotti F. PCL LOF Filter Implementation. 2018 [cited 2018 Jan 1]. Available from: https://github.com/fpirotti/PCL-LOFFilter
29. Rusu RB, Cousins S. 3D is here: point cloud library (PCL). Shanghai, China: IEEE Int. Conf. Robot. Autom; 2011.
30. Muja M, Lowe DG. Fast Matching of Binary Features. In: Compututer and Robot Vision (CRV); 2012. p. 404–10.
31. Muja M, Lowe DG. Scalable Nearest Neighbor Algorithms for High Dimensional Data. IEEE Trans Pattern Anal Mach Intell 2014;36(11):2227-2240
32. Isenburg M. LASlib (with LASzip). 2017. Available from: https://github.com/LAStools/LAStools/tree/master/LASlib. Accessed 01 Jan 2018.
33. SQLite library [Internet]. 2018. Available from: https://www.sqlite.org/about.html. Accessed 01 Jan 2018.
34. Sachs MC. Generate ROC Curve charts for print and interactive use [internet]. 2017. Available from: https://cran.r-project.org/web/packages/plotROC/. Accessed 01 Jan 2018.
35. Fawcett T. ROC Graphs : notes and practical considerations for researchers. ReCALL. 2004;31:1–38. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.9777&amp;rep=rep1&amp;type=pdf
36. Ling CX, Huang J, Zhang H. AUC: A statistically consistent and more discriminating measure than accuracy. Int Jt Conf Artif Intell. 2003:519–24.