Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data

Abdul Nurunnabi [a,*], Geoff West [b], David Belton [b]

[a, b]Department of Spatial Sciences, Curtin University, Perth, Western Australia-6845, Australia
[a]abdul.nurunnabi@postgrad.curtin.edu.au, [b]{ G.West, D.Belton }@curtin.edu.au
[a,b]Cooperative Research Centre for Spatial Information

Abstract—This paper proposes two robust statistical techniques for outlier detection and robust saliency features, such as surface normal and curvature, estimation in laser scanning 3D point cloud data. One is based on a robust z-score and the other uses a Mahalanobis type robust distance. The methods couple the ideas of point to plane orthogonal distance and local surface point consistency to get Maximum Consistency with Minimum Distance (MCMD). The methods estimates the best-fit-plane based on most probable outlier free, and most consistent, points set in a local neighbourhood. Then the normal and curvature from the best-fit-plane will be highly robust to noise and outliers. Experiments are performed to show the performance of the algorithms compared to several existing well-known methods (from computer vision, data mining, machine learning and statistics) using synthetic and real laser scanning datasets of complex (planar and non-planar) objects. Results for plane fitting, denoising, sharp feature preserving and segmentation are significantly improved. The algorithms are demonstrated to be significantly faster, more accurate and robust. Quantitatively, for a sample size of 50 with 20% outliers the proposed MCMD_Z is approximately 5, 15 and 98 times faster than the existing methods: uLSIF, RANSAC and RPCA, respectively. The proposed MCMD_MD method can tolerate 75% clustered outliers, whereas, RANSAC and RPCA can only tolerate 47% and 64% outliers respectively. For outlier detection, with a data size of 50 with 20% outliers, MCMD_Z has an accuracy of 99.72% , 0.4% false positive rate and 0% false negative rate; for RPCA, RANSAC and uLSIF, the accuracies are 97.05%, 47.06% and 94.54% respectively, and they have misclassification rates higher than the proposed methods. The new methods have potential for local surface reconstruction, fitting, and other point cloud processing tasks.

Keywords- feature extraction; outlier detection; plane fitting; point cloud denoising; robust curvature; robust normal; saliency feature; segmentation; surface reconstruction

1. Introduction

Many algorithms for point cloud processing use information about local saliencies, such as surface normal and curvature, at each point of the data. Reliable surface reconstruction, object modelling and rendering

[*] Corresponding author: Tel.: +61 8 92662704; Fax: +61 8 92931673.

applications heavily depend on how well the estimated local surface normals and curvatures approximate the true normals and curvatures of the scanned surface [1,2]. Many studies on accurate normal and curvature estimation have been carried out over the years in computer graphics, computer vision, pattern recognition, photogrammetry, reverse engineering, remote sensing and robotics [2-7].

One of the first attempts by Hoppe et al. [3] for normal and curvature estimation assumed that the underlying surface is locally smooth throughout the entirety of the data. This assumption has the advantage of approximating the local neighbourhood of a given point by a planar surface. Following the work of Hoppe et al. [3], PCA based local saliency features have been a focus for point cloud processing including plane fitting, feature extraction, surface segmentation and reconstruction [8-10] mainly because of its simplicity and speed. However, it has been shown that PCA is sensitive to outliers and fails to reliably fit planar surfaces in their presence, therefore saliency features based on PCA are not robust and the resultant analyses can be erroneous and misleading [11,12]. In addition, in the vicinity of geometric singularities (e.g. corners or edges) where the normals are discontinuous, PCA fails to preserve sharp features since neighbouring points are used non-distinctively to compute the planar fit. The effect is smoothed normal estimates along the edges [13]. As such, there is a great interest in applying robust and accurate methods in presence of outliers efficiently.

The word ‘outlier’ has been defined by many ways depending on the applications. Simply, an outlier is an observation that is (a) so far from the majority of observations, or (b) somehow can be differentiated by the general behaviour (pattern) of the majority, that it should be treated differently [14,15]. The presence of outliers or gross errors is an unavoidable phenomenon in point cloud data as it is one of the main problems facing accurate normal and curvature estimation. Outliers occur mainly because of the physical limitations of the data collection sensors, discontinuities at boundaries between 3D features, occlusions, multiple reflectance, and noise that produces off-surface points [7]. Outlier detection in point cloud data becomes complex because the points are usually unorganized, noisy, sparse, inconsistent in point density, have geometrical discontinuities, arbitrary surface shape with sharp features, and there is little to no knowledge about the theoretical statistical distribution of the points. Besides, it is common to get multiple model structures in the data that can create clustered outliers to one structure of interest but inliers to another structure, e.g. a pole in front of a flat wall, may appear as pseudo outliers. Fleishman et al. [16] stated that when the underlying surface contains sharp features, the requirement of being resilient to noise is especially challenging since noise and sharp features are ambiguous, and most techniques tend to smooth important sharp features. Despite recent progress in pattern recognition, statistics, computer vision, data mining and machine learning techniques for processing point data, the problem of outlier (specially clustered outliers) detection in unstructured point clouds is still a challenging task [7, 14, 17-20].

This paper proposes two robust outlier detection algorithms that can identify a large percentage of clustered outliers as well as uniform outliers. The outlier detection algorithms couples with PCA used to estimate robust local saliency features such as normals and curvature. The key idea is to use local neighbourhood information instead of global information of the data, assuming that in a certain sufficiently small local neighbourhood, the points are on a planar surface [3]. The proposed algorithms have the two following stages:

- First, outliers and/or noise are identified in a local neighbourhood for every point in the data. The outliers are identified by using robust statistical approaches based on measures of the distance of a point to the plane (based on its local neighbours) and the local surface point variation along the normal.
- Second, the best-fit-plane and the relevant parameters such as normal and curvature, are estimated using PCA after the outlying cases found by the first stage have been removed.

We compare the results of the proposed methods with the statistical methods PCA [21] and robust PCA [22], computer vision methods RANSAC [23, 24] and MSAC (M-estimator SAmple Consensus) [24, 25], machine learning method uLSIF; [26, 27], and data mining method qs_p; [28]. The accuracy, robustness and speed of the computation of the methods are compared with respect to size of the data, outlier percentage, and point density variation. We also evaluate the saliency features estimated using the new methods for different applications including point cloud denoising, sharp feature preserving and segmentation.

The remainder of the paper is organized as follows. Relevant literature is reviewed in Section 2. In Section 3, the related principles and methods of outlier detection in statistics, computer vision, machine learning and data mining are briefly discussed. Section 4 proposes two variants of outlier detection and saliency feature estimation algorithms. In Section 5, the algorithms are demonstrated and evaluated through comparison with the established techniques mentioned above using simulated and real laser point cloud data. Section 6 concludes the paper.

2. Literature review

This paper proposes methods for outlier detection and determining robust normal and curvature. The relevant literature is reviewed in two sections: (i) robust normal and curvature estimation, and (ii) outlier detection.

2.1 Robust normal and curvature estimation methods

A number of methods have been developed to improve the quality and speed of normal and curvature estimation in point cloud data. Methods are proposed and tailored according to their suitability for the particular application e.g. plane fitting [29, 30], surface reconstruction [3, 31], sharp feature preserving [6,16] and normal estimation [11,32,33].

Algorithms for normal estimation can be categorized into two major approaches: combinatorial and numerical approaches [2,13]. The first approach is based on the information extracted from Delaunay and Voronoi properties [2,32]. Dey et al. [2] developed combinatorial methods for estimating normals in the presence of noise, but in their comparative study it is shown that in general, this approach becomes infeasible for large datasets. Numerical approaches find a subset of points in the local neighbourhood that may represent the local surface of an interest point and is known to perform better in the presence of outliers and noise. Then the best-fit-plane to the selected subset is computed and the normal of the plane is treated as the estimated normal for the point of interest. Hoppe et al. [3] estimated the normal at each point to the fitted plane of the nearest neighbours by applying regression. The total least squares is regarded as a numerical approach that is computed efficiently by PCA. PCA based plane fitting is also known as *PlanePCA* [4], which is a geometric optimization which can be shown to be equivalent to the Maximum Likelihood Estimation (MLE) method [29]. Klasing et al. [4] compared a number of optimization and averaging methods and concluded their paper by stating that in the case in which a *k*-Nearest Neighbour (*k*-NN) graph is maintained and updated, the *PlanePCA* is the universal method of choice because of its superior performance in terms of both quality and speed. It is known that the PCA based method minimizes the LS cost function. Hence, the results from PCA are affected by outlying observations because of the covariance matrix used here has an unbounded 'influence function' and a zero 'breakdown point' [19,22]. Distance weighting [34], changing neighbourhood size [11] and higher-order fitting [9] algorithms have been developed to adjust PCA for better accuracy near sharp features. Castillo et al. [13] claimed that such improvements in PCA fail to address the fundamental problem of determining which points contained in a given neighbourhood should contribute to the normal estimation. Fleishman et al. [16] proposed a forward search approach based robust moving least squares technique for reconstructing a piecewise smooth surface and reliable normal estimation. The method can deal with multiple outliers but requires very dense sampling and a robust initial estimator to start the forward search algorithm. Sheung and Wang [31] showed that the forward search misclassifies the region when it fails to obtain a good initial fit. Hence the resultant estimates may be erroneous. Oztireli et al. [35] used local kernel regression to reconstruct sharp features. Weber et al. [6] claimed the reconstruction from Oztireli et al. [35] does not have a tangent plane at a discontinuous sharp feature, but only gives the visual effect of a sharp feature during rendering. There are two solutions developed for handling outliers in the literature: (i) outlier detection and (ii) robust methods. The following section gives a summary of existing outlier detection and robust methods.

2.2 Outlier detection and robust methods

Most of the work that has been performed for outlier detection exist in statistics. However, many outlier detection approaches have been developed in machine learning, pattern recognition and data mining and are referred to by different names e.g. novelty detection, anomaly detection, exception mining or one-class classification. These also depend on application areas, which include information systems, network systems, news documentation, industrial machines, and video surveillance [17, 18, 20, 27, 36-39]. Existing methods can be broadly arranged into four groups as follows. First, statistical methods: broadly categorised into distribution and depth based methods, where outliers are identified based on standard probability distributions that fit the data best, and in a *k*-dimensional space assigning a depth, respectively [14, 19]. One of the main limitations of distribution based approaches is that the information about the underlying data distribution may not always be available. The second type of outlier detection method is distance and/or density based methods: Knorr and Ng [37] generalize the distribution based approach and formulated the notion of Distance Based (DB) outlier detection for large data. In contrast to DB methods that take a global view of the data, Breunig et al. [36] introduced a density based approach assuming that objects may be outlier relative to their local neighbourhood. Distance and density based approaches triggered interest in the development of many variants of the algorithms, which are more spatially oriented [20, 28, 40]. Thirdly, clustering based methods apply unsupervised clustering techniques mainly to group the data based on their local data behaviour [41]. Small clusters that contain significantly less data points are identified as outliers. The performance of the clustering based methods is highly sensitive to the clustering techniques that are involved in capturing the cluster structure of the normal (inlier) data [39]. The last approach is the model based approach that is used to learn a model (classifier) from a set of known data, i.e. training data, and then classifies test observations as either normal or outlier using the learnt model [38, 39, 42]. In this category, Tax and Duin [42] introduced Support Vector Data Description (SVDD). Usually, model based approaches can detect outliers in high-dimensional data but require much more time to construct a classifier [39]. Hido et al. [26] pointed that the solutions of the One-class Support Vector Machine (OSVM) and SVDD depend heavily on the choice of the tunning parameters and there seems to be no reasonable method to appropriately fix the values of the tuning parameters. Several survey papers [15, 18, 20, 43] have been published in the last decade that explored a variety of algorithms covering the full range of statistics, machine learning and data mining techniques. Hodges and Austin [18] concluded: there is no single universally applicable or generic outlier detection approach.

Robust approaches have been developed to avoid (or reduce) the outlier/noise influence on the estimates. Many robust versions of PCA have been introduced in the statistical literature [22, 44]. Nurunnabi et al. [12] used fast-MCD [45] based robust PCA [22] for planar surface fitting. The RANSAC [23] paradigm introduced in computer vision is a model-based algorithm known as a robust technique, that is very often used in laser scanning for planar surface detecting, fitting, extraction and normal estimation [46]. Deschaud and Goulette [30] showed that RANSAC is very efficient at detecting large planes in noisy point clouds.

3. Related principles and methods

This section briefly describes the basic principles and the relevant methods used for comparison.

3.1 PCA and local covariance statistics

PCA is a statistical technique that explains the covariance structure of the data by means of a small number of variables called Principal Components (PCs) [21]. PCs are the linear combinations of the mean centred original variables that rank the variability to the orthogonal directions. The first PC shows the direction in which the projected observations have the largest variance; the second PC shows the direction of the next largest variance not explained by the first PC, and so on. Based on the information of the local neighbourhood of a point $p_i = (x_i,\ y_i,\ z_i)$, where p_i is a 3D point in a point cloud P, the covariance matrix Σ is defined as:

$$\Sigma_{3\times 3} = \frac{1}{k}\sum_{i=1}^{k}(p_i - \bar{p})\,(p_i - \bar{p})^T, \tag{1}$$

where $\bar{p} = \frac{1}{k}\sum_{i=1}^{k}(p_{x_i},\ p_{y_i},\ p_{z_i})$ is the mean of the k-size neighbourhood Np_i. The covariance matrix Σ in Eq. (1) is able to define the geometric information of the underlying local surface. Using Singular Value Decomposition (SVD), Σ is decomposed into eigenvectors (PCs) usually sorted in descending order of the eigenvalues λ_i $(i = 2,1,0)$. The eigenvalues λ_2, λ_1, and λ_0 correspond to the eigenvectors v_2, v_1, and v_0 , respectively. For a sampled surface, the first two PCs explain most of the variability, and they are enough to define a locally planar surface in 3D. For a set of 3D points, a planar equation is defined as:

$$ax + by + cz + d = 0, \tag{2}$$

where a, b and c are the slope parameters and d is the distance of the plane to the origin. Thus v_o approximates the surface normal $(\hat{n})$ for the point p_i [3], and the elements of the eigenvector v_o are the estimated plane parameters. The least eigenvalue λ_0 describes the variation along the surface normal and can measure how the points are consistent among themselves. Pauly et al. [8] used the eigenvalues to get surface variation along the direction of the corresponding eigenvalues and defined surface variation as the curvature at p_i as:

$$\sigma(p_i) = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}, \quad \lambda_2 > \lambda_1 > \lambda_0\ . \tag{3}$$

Unfortunately, both the classical variance and the classical covariance matrix, which is being decomposed, are very sensitive to outliers. As a result, attributes calculated through PCA are highly sensitive to outlying points [22].

3.2. Methods used for comparison

We briefly discuss the methods that can make PCA results robust. We consider several robust and outlier detection methods from different disciplines, which are popular and/or have been considered state-of-the-art.

3.2.1. Robust principal component cnalysis

The aim of robust PCA (RPCA) is to obtain the PCs that are not influenced by outliers/noise in the data. Many versions of RPCA have been developed based on the appropriateness to the number of dimensions of the data [44, 47]. We are dealing with 3D point cloud data, so the low dimensional RPCA methods are considered. These types of methods can be categorized mainly into: robust covariance based and Projection Pursuit (PP) based [48]. Hubert et al. [22] combined the ideas of both robust covariance estimation and PP, and proposed a RPCA. They claimed that their method yields accurate estimates for outlier free datasets and more robust estimates for outlier contaminated data, and has the further advantage of outlier detection.

The RPCA [22] algorithm is performed as follows. First, the PP technique is used to make sure that the transformed data are lying in a subspace whose dimension (*m*) is less than the number (n) of observations. Second, an 'outlyingness' measure is computed by projecting all the data points onto many univariate directions. To reduce computation time the dataset is compressed to PCs defining potential directions. Then, every direction is scored by the Stahel-Donoho outlyingness measure defined as in [15, 22]:

$$w_i = \arg\max_v \frac{|p_i v^T - c_{FMCD}(p_i v^T)|}{\Sigma_{FMCD}(p_i v^T)}, \qquad i = 1, \dots, n \tag{4}$$

where $p_i v^T$ denotes a projection of the i^{th} observation onto the *v* direction, c_{FMCD} and Σ_{FMCD} are the fast-MCD [45] based mean and covariance matrix for the univariate direction *v*. An assumed portion h ($h > n/2$) of observations with the smallest outlyingness values is then used to construct a robust covariance matrix Σ_h. A larger *h* can give a more accurate RPCA but a smaller *h* gives more robust results. We choose $h = \lceil 0.5n \rceil$ unless the data is contaminated with more than 50% outliers. Finally, the PCA model is built on the robust Σ_h. Based on the PCA model, the remaining observations are projected onto the *d* dimensional subspace spanned by the *d* largest eigenvectors of Σ_h. The RPCA algorithm is able to find and score orthogonal outliers. An orthogonal outlier is identified by a large orthogonal distance which is the distance between the observation *p* and its projection $\hat{p}$. A scored outlier is identified by the score distance which is separated in the PCA subspace. The reader is referred to [22] for full details on the RPCA algorithm.

3.3.2. RANSAC and MSAC

RANSAC [23] is a well-known robust method developed in computer vision. This has been used for robust parameter estimation for a given model. The iterative algorithm is composed of two steps: Hypothesize and Test. In Hypothesize, a minimal subset, e.g. three points for a plane, is randomly selected and the required parameters are computed based on the subset. In Test, the estimates are tested for support from the whole set of putative correspondences, i.e. the consensus set. The support is the number of correspondences or inlier candidates with error below a predefined threshold. RANSAC finally chooses the most probable hypothesis from a number of iterations supported by the most inliers evaluated by a Least Squares (LS) cost function. RANSAC can fit the model with a high percentage of outliers, but breaks down when the error thresholds are incorrectly defined regardless of the number of outliers, and in the presence of multiple models [25, 49]. Torr and Zisserman [25] adopted a bounded cost function, an M-estimator, and proposed MSAC [25], which minimizes the robust cost function $C_f = \sum_i \rho(e_i^2)$, where e is the error term and $\rho\ (e^2)$ is defined as:

$$\rho(e^2) = \begin{cases} e^2 & e^2 < T^2 \\ T^2 & e^2 \geq T^2, \end{cases} \tag{5}$$

and where inliers are scored according to their goodness-of-fit to the data. The authors claim that MSAC yields a modest to hefty benefit for all robust estimations with absolutely no additional computational burden. Although a probabilistic iteration number [19, 23] is used in RANSAC to ensure the likelihood of at least one outlier-free model, it has been argued that an outlier-free model estimated from all inliers does not guarantee a good solution due to noise disturbance and others phenomena [50]. Methods have been proposed to update the number of iterations [24, 50]. We use the MATLAB® code of Zuliani [24] for performing RANSAC and MSAC algorithms which adopts the developments in [24, 50] and gives better results in the presence of noise.

3.3.3. Direct density ratio based method

A well-known method in the statistical and machine learning literature performs outlier detection using a density ratio based approach which takes the ratio of the two probability density functions of test and training datasets. The approach for identifying outliers in a test or validation dataset based on a training or model dataset that only contains regular or inlier data [27, 38]. Density estimation is not trivial and getting an appropriate parametric model may not be possible. Therefore, Direct Density Ratio (DDR) estimation methods have been developed without going through the density estimation. Recently, Hido et al. [26] introduced an inlier based outlier detection method based on DDR estimation that calculates an inlier score or importance defined as:

$$w(p) = \frac{p_{tr}(p)}{p_{te}(p)}, \tag{6}$$

where $p_{tr}(p)$ and $p_{te}(p)$ are the densities of identically and independently distributed (i.i.d.) training $\{p_j{}^{tr}\}_{j=1}^{n_{tr}}$ and test $\{p_i{}^{te}\}_{i=1}^{n_{te}}$ samples respectively. The observations with small inlier scores are potentially outliers. The authors [26] use unconstrained Least Squares Importance Fitting (uLSIF), which originated from the idea of LSIF [27]. In uLSIF, the closed-form solution is computed by solving a system of linear equations. The importance $w(p)$ in Eq. (6) is modelled as:

$$\widehat{w}(p) = \sum_{l=1}^{b} \alpha_l \, \varphi_l(p), \tag{7}$$

where $\{\alpha_l\}_{l=1}^{b}$are parameters and $\{\varphi_l(p)\}_{l=1}^{b}$are basis functions such that $\varphi_l(p) \geq 0$. The parameters are determined by minimizing the following objective function:

$$\frac{1}{2}\int \left(\widehat{w}(p) - \frac{p_{tr}(p)}{p_{te}(p)}\right)^2 p_{te}\,(p)dx. \tag{8}$$

The solution of uLSIF is computed through matrix inversion, and the leave-one-out-cross-validation score [27] for uLSIF computed analytically. Hido et al. [26] showed that the uLSIF is competitively accurate and computationally more efficient than the existing best methods e.g. OSVM [38] and Local Outlier Factor (LOF) [36]. The reader is referred to Hido et al. [26] for further details about uLSIF.

3.3.4. Distance based outlier detection

Knorr and Ng [37] first introduced the new paradigm of Distance Based (DB) outlier detection that generalises the statistical distribution based approaches. In contrast to statistical distribution based approaches, it does not need prior knowledge about the data distribution. In DB outlier detection, a point p is considered as an outlier w.r.t. parameters α, δ if at least a fraction α of data has a distance from p larger than δ, that is:

$$|\{q \in P | d(p,q) > \delta\}| \geq \alpha n, \tag{9}$$

where $q \in P$, and $(\alpha, \delta) \in \mathbb{R}$; and $0 \leq \alpha \leq 1$ are the user defined parameters. But the problem is to fix the distance threshold δ. Ramaswamy et al. [51] proposed k^{th}Nearest Neighbour (k^{th}NN) distance as a measure of outlyingness to overcome the limitation. The score of a point is defined as:

$$q_{k^{th}\mathrm{NN}}(p) := d^k(p; P), \tag{10}$$

where $d^k(p;P)$ is the distance between p and its k^{th}NN. Since this method is computationally intensive, Wu and Jermaine [52] proposed a sampling algorithm to efficiently estimate the score in Eq. (10), defined as:

$$q_{k^{th}\mathrm{S}_p}(p) := d^k(p, S_p(P)), \tag{11}$$

where $S_p(P)$ is a subset of P, which is randomly and iteratively sampled for each point in P. To save computation time without losing the accuracy, recently Sugiyama and Borgwardt [28] suggested sampling only once. They define the score as:

$$q_{S_p}(p) := \min_{q \in s_p} d(p, q), \qquad (12)$$

where $\min_{q \in s_p} d(p, q)$ is the minimum distance between p and q, where q is a point in the subset s_p. The authors name the algorithm q_{S_p}, and state that it outperforms state-of-the-art DB algorithms including LOF [36], Angle Based Outlier Factor (ABOF) [53] and OSVM [38] in terms of efficiency and effectiveness.

4. Proposed methods for outlier detection and robust saliency features estimation

This section proposes two algorithms for outlier detection and robust saliency feature estimation in point cloud data. The algorithms perform four sequential tasks shown in the following diagram (Fig. 1).

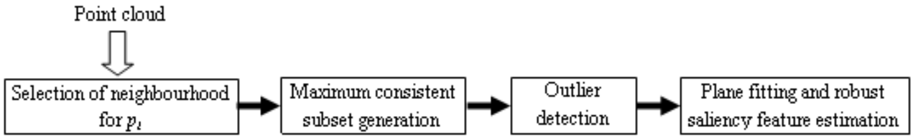

Point cloud
Selection of neighbourhood for p_i
Maximum consistent subset generation
Outlier detection
Plane fitting and robust saliency feature estimation

Fig. 1. Outlier detection and robust saliency features estimation process.

It is known that finding outliers globally in a point cloud is not appropriate because of the presence of multiple object surfaces, as well as clustered and/or pseudo-outliers. So the aim of the proposed algorithms is to find outliers locally. We find outliers for each and every point within their local neighbourhood to get the benefits from the fact that an outlier free local neighbourhood will produce more accurate and robust local saliency features. In the case of neighbourhood based point cloud processing, it can be assumed that within a local neighbourhood of appropriate size, data points can be assumed to be sampled from a locally planar surface. Therefore, we need to find a local region (surface) for an interest point p_i by searching its local neighbourhood. The two well-known neighbourhood searching methods in point cloud analysis are the Fixed Distance Neighbourhood (FDN) and the k-Nearest Neighbourhood (k-NN). The FDN method selects all points within a fixed radius *r* around p_i, whereas *k*-NN finds the *k* points having the least distance from p_i. We prefer *k*-NN, because it can avoid the problem of point density variation and lack of adequate number of redundant observations. We know point density variation is a common event when dealing with Mobile Laser Scanning (MLS) data because of the variation in movement and the geometry of the data acquisition sensors or vehicles. In addition to that, this type of local neighbourhood can produce local statistics e.g. normals with support of the same number of points.

We follow the basic philosophy of diagnostic statistics, which proceeds with identifying outliers, removes them and then fitting to the remaining data in the classical way [19]. Hence, after finding outliers in a local neighbourhood, we fit the plane to the outlier free data using classical PCA, and estimate the local saliencies, that is normal and curvature, using the best-fit-plane parameters and estimated eigenvalues. The algorithms serve two purposes: (i) outlier detection in the point cloud data, and (ii) robust saliency feature estimation. Outlier detection can also be used in point cloud denoising, as illustrated in Section 5.2.1. Estimated robust saliency features can be used for local neighbourhood based point cloud processing such as sharp feature preserving and segmentation, which is presented in Sections 5.2.2 and 5.2.3.

The two proposed robust outlier detection methods use the robust *z* or *Rz*-score and robust Mahalanobis Distance (RMD). To identify outliers, the algorithms couple the idea of using point to plane Orthogonal Distances (OD) and the surface points variation λ_o along the normal. Only the h-subset (a subset of size *h*) of the majority of good points in a local neighbourhood that are most reliable, homogenous and have the minimum sorted ODs are used to fit the plane and to calculate respective λ_o values. The decision based on the majority of consistent points is a fundamental idea in robust statistics [19]. Fixing *h* removes the problem of choosing an explicit value of the error threshold that is a major problem in the RANSAC paradigm [23]. In general, we set $h = \lceil 0.5k \rceil$ to get the majority of consistent points. In order to get the best *h*-subset, the algorithm starts with a random h_o-subset, where the h_o-subset has the minimal number of points. In the case of plane fitting, $h_o = 3$. The technique of finding the *h*-subset by using the h_o-subset reduces the iteration time, because h_o is considerably smaller than *h*. Using the outlier-free minimal subset (MS) the *h*-subset can produce better plane parameters. Consequently it gives a better and more accurate normal and the relevant error scale, or point to plane orthogonal distance, for the most consistent h-subset, which is used to get the best-fit-plane and robust saliency features. To get an outlier free h_o-minimal subset, one could iterate by randomly sampling ${}^{k}C_{h_o}$ times, where *C* mean combination, but the number of iterations increases rapidly with the increase of k. We employ a Monte Carlo type probabilistic approach [19] to calculate the number of iterations I_t. Given the outlier rate ϵ which is the probability that a point is an outlier, if we set P_r for the desired probability that at least one outlier free h_o-subset can be found from the percentage ϵ of outlier contaminated data, then $P_r = 1 - (1 - (1 - \epsilon)^{h_o})^{I_t}$, and I_t can be defined as:

$$I_t = \frac{\log(1-P_r)}{\log\left(1-(1-\epsilon)^{h_o}\right)}. \qquad (13)$$

Therefore, $I_t = f(P_r, \epsilon, h_o)$, where $h_0 = 3$ is fixed. In this paper, we use P_r = 0.9999 although users have the freedom to choose P_r based on their knowledge about their data. Fixing a larger probability increases the number of iterations giving a more accurate, more consistent subset with a high probability of the subset being outlier free. It is known that the number of iterations is a trade-off between accuracy and efficiency. The outlier rate ϵ is generally unknown *a priori*. A smaller ϵ than the real outlier percentage in the data can be influenced by the masking effect. However, an excessively large value of ϵ can create swamping. Masking occurs when an outlying case is unidentified and misclassified as a good one, and swamping occurs when regular observations are incorrectly identified as outliers [15, 43]. Experience of MLS data reveals that generally, the majority or more than 50% of points are inliers within a local neighbourhood. To keep the computation safe, we assume $\epsilon = 0.5$ for real data. The user can change ϵ based on knowledge about the presence of outliers. We find an h-subset for every iteration, based on the minimum orthogonal distance with respective to the corresponding fitted plane of the h_o-subset, and calculate $\lambda_o$values for all the h-subsets from the I_t iterations. It is reasonable to assume that the plane w.r.t. to the least λ_o value also has maximum surface consistency, that is the least variation along the normal, among all the h-subsets. Theoretically, the maximum consistency is attained at $\lambda_o \approx 0$. In this way, we get maximum surface point consistency from the points that have minimum ODs to the fitted plane. We dub the method Maximum Consistency with Minimum Distance (MCMD).

The algorithms for robust outlier detection and saliency feature estimation can be summarized and organized as described in the following three subsections.

4.1. Getting the maximum consistent set

The proposed outlier detection algorithms are inspired by the concept of robust outlier detection in statistics: detecting the outliers by searching for the model fitted by the majority of the data [15, 19]. The subset of majority points used in our algorithms includes the most homogenous and consistent points w.r.t. to each other. The Maximum Consistent Set (MCS) can be derived by the following steps in Algorithm 1.

Algorithm 1: MCS

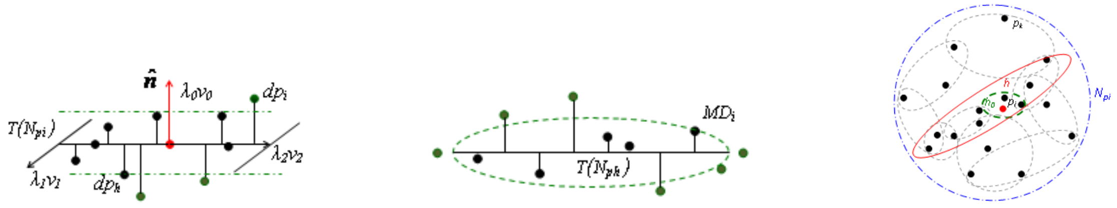
Step 1. To get the h-subset for the MCS in a local neighbourhood of a point of interest p_i, we randomly choose h_o points, in our case $h_o = 3$, to fit a plane. If the rank of this subset is less than h_o, randomly add more points gradually to the subset until the rank is equal to h_o.

Step 2. For the above h_o-subset, we fit a plane by PCA and calculate ODs for all the points in the local neighbourhood to the fitted plane and sort them according to their ODs (Fig. 2a) as: $|OD(p_1)| \leq, \ldots, \leq |OD(p_h)|, \ldots, \leq |OD(p_k)|$,
where $OD(p_i) = (p_i - \bar{p})^T . \hat{n}$ is the OD for the point p_i to the fitted plane, and $\bar{p}$ and $\hat{n}$ are the mean and the unit normal of the fitted plane, respectively.

Step 3. Fit the plane to the above sorted h-subset in Step 2, and calculate the λ_0 value for that plane and add it to the list of previous λ_o values, defined as $S(\lambda_o)$.

Step 4. Iterate Step 1 to Step 3 I_t times as defined in Eq. (13). We add the λ_0 values for the I_t times in $S(\lambda_o)$.

Step 5. Find the h-subset of points for which λ_0 is minimum in $S(\lambda_o)$. This is the required MCS (red ellipse in Fig. 2c) in the local neighbourhood (Np_i) of a point p_i.



(a) (b) (c)

Fig. 2. (a) Point to plane orthogonal distance $OD(p_i)$ or dp_i (b) robust MD ellipse, and (c) blue dotted big circle is a local neighbourhood Np_i, black dotted circles/ellipses are h_o-subsets, red ellipse is the MCS, and the green dash-dotted ellipse is the h_o-subset w.r.t. the MCS which produces the least λ_o value.

4.2. *Outlier detection*

We employ an outlier detection method from statistics [14, 15, 19]. Outlier detection methods can be distinguished depending on the number of dimensions of the data. One of the simplest methods in statistics is the distance-based approach using the data distribution that aims to find outliers by computing the distances of the points in a dataset from its majority (centre), and a point that is significantly far from the centre of the data treated as an outlier.

In the univariate case, the well-known z-score is a distance-based measure that can be defined as the standardized residual:

$$z_i = \frac{|p_i - \bar{p}|}{\sigma_p}, \quad i = 1, \ldots, k \tag{14}$$

where $\bar{p}$ and σ_p are the mean and standard deviation (StD) of the variable P. Although the z-score is very simple and easy to compute, the inclusion of the mean and StD makes its Breakdown Point (BP) zero [19]. That means a single outlier can move the mean and scatter arbitrarily far from the real values. The most popular robust alternatives to the mean and standard deviation are the median and Median Absolute Deviation (MAD) respectively, both of which have the best possible BP of 50%, where $\text{MAD} = a.\text{median}_i\ |p_i - \text{median}_j(p_j)\ |$; with a =1.4826 set as a correction factor used to make the estimator consistent [15]. To reduce the outlier sensitivity of the *z*-score, the median and MAD are utilised in place of the mean and StD in Eq. (14) to produce the robust *z*-score (*Rz*):

$$Rz_i = \frac{|p_i - \text{median}_j(p_j)|}{\text{MAD }(p)}, \qquad i = 1, \dots, k \tag{15}$$

which is much more reliable and robust than the *z*-score. Observations with z_i or Rz_i values greater than 2.5 are considered as outliers [15].

In the case of multivariate outlier detection, the scatter of the data is as equally important as the centre. One of the most well-known multivariate outlier detection methods is the Mahalanobis Distance (MD), defined as:

$$\text{MD}_i = \sqrt{(p_i - \bar{p})^T \Sigma^{-1} (p_i - \bar{p})}, \quad i = 1, \dots, k \tag{16}$$

where $\bar{p}$ and Σ are the sample mean and covariance matrix. MD follows a Chi-square (χ^2) distribution with the number of degrees of freedom equal to the number of variables *m,* and the observations that exceed $\surd(\chi^2_{m,0.975})$ = 3.075 are identified as outliers. The non-robust mean and covariance matrix influence MD, and the results no longer suffice for multiple and clustered outliers because of masking and swamping effects [45]. Hence, obtaining robust estimators of mean and covariance matrix is a precondition for a robust MD type outlier detection method.

The proposed algorithms identify outliers in a local neighbourhood of p_i in two ways: (i) using the robust *z*-score, and (ii) using the Robust Mahalanobis Distance (RMD). The methods are dubbed as MCMD_Z and MCMD_MD, respectively. The two outlier detection algorithms are summarized in the following algorithms.

```
Algorithm 2. MCMD_Z
Input: $Np_i$: Neighbourhood of point $p_i$,
MCS: Maximum consistent set.
Output: $INdx$: Inlier indices of $Np_i$,
$OINdx$: Outlier indices of $Np_i$.
1 Fit the plane using the $h$-MCS from Algorithm 1
  and estimate the plane normal $\hat{n}_h$
2 Calculate the robust mean $\bar{p}_h$ from MCS:
  $\bar{p}_h = \frac{1}{h}\sum_{i=1}^{h}(p_x, p_y, p_z)$
```

```
Calculate robust ODs (as shown in Fig. 2a) for all points $Np_i$:
    $OD(p_i) = (p_i - \bar{p}_h)^T . \hat{n}_h, \quad i = 1,2,\dots,k$
Calculate the Rz-score for all points using the ODs:
    $Rz_i = \frac{|OD_i - \text{median}_j(OD_j)|}{MAD(OD)}, \quad i = 1,2,\dots,k \quad (17)$
for $i = 1$ to $k$ do
    If $Rz_i < 2.5$ then
        $INdx \leftarrow i$
    else
        $OINdx \leftarrow i$
    end if
end for
Return: $INdx$ and $OINdx$.
```

Algorithm 3. MCMD_MD

Input: Np_i: Neighbourhood of point p_i,
MCS: Minimum consistent set.

Output: $INdx$: Inlier indices of Np_i,
$OINdx$: Outlier indices of Np_i.

```
Calculate the robust mean $\bar{p}_h$ and covariance matrix $\Sigma_h^{-1}$ from MCS: $\bar{p}_h = \frac{1}{h}\sum_{i=1}^{h}(p_x, p_y, p_z)$
Calculate the robust MDs (as shown in Fig. 2b) for all points as:
    $\text{RMD}_i = \sqrt{(p_i - \bar{p}_h)^T \Sigma_h^{-1} (p_i - \bar{p}_h)}, i = 1,.,k \quad (18)$
for $i = 1$ to $k$ do
    If $\text{RMD}_i < 3.075$ then
        $INdx \leftarrow i$
    else
        $OINdx \leftarrow i$
    end if
end for
Return: $INdx$ and $OINdx$.
```

4.3. *Robust saliency features estimation*

By removing the outliers, we get an outlier free neighbourhood for the *i*th point p_i as described in Section 4.2. We now fit the plane without the outliers using PCA. Estimated eigenvalues and eigenvectors are used to get the required robust saliency features normal and curvature. The least eigenvector, that is the third PC, is used as the robust normal $\hat{n}$, and the surface variation defined in Eq. (3) is known as the robust curvature σ_p [8]. The algorithm for estimating the robust saliency features normal and curvature is as follows.

Algorithm 4. Robust saliency features estimation

Input: P: Point cloud, k: neighbourhood size.

Output: $\{\sigma(p)\}$: set of curvature values, $\{\hat{n}\}$: set of normals, and $\{\lambda_i\}$: set of eigenvalues.

```
for $i = 1$ to $n$ do
    Find the k-nearest neighbourhood $Np_i$ for point $p_i$
    Find the MCS in $Np_i$ using Algorithm 1
    Remove outliers using Algorithm 2 or Algorithm 3 from $Np_i$
    Perform classical PCA on the cleaned $Np_i$
    Arrange the three PCs associated with their respective eigenvalues
    Find the two PCs with the largest eigenvalues that form the basis for the fitted plane
    The 3rd PC with the smallest eigenvalue can be used as the normal $\hat{n}_i$ of the fitted plane
    Calculate $\sigma(p_i)$ using Eq. (3)
end for
Return: $\{\sigma(p)\}$, $\{\hat{n}\}$ and $\{\lambda_i\}$.
```

5. Experiments and evaluation

The proposed algorithms are demonstrated and evaluated in terms of accuracy, robustness, breakdown points, classification into outliers and inliers, and speed of computation using synthetic and real MLS datasets. Estimated local saliencies of normal and curvature values are evaluated for point cloud denoising, sharp feature preserving and segmentation of 3D point clouds. We compare our methods (MCMD_Z and MCMD_MD) with PCA, RANSAC, MSAC, RPCA, uLSIF and q_{S_p}.

To evaluate the performance, we fit the planar surface for a local neighbourhood Np_i of an interest point p_i using the different methods, estimate normal and eigenvalue characteristics λ_o and σ_p, and use them for point cloud processing. We calculate three measures: (i) the bias or dihedral angle θ between the planes fitted to the local neighbourhood with and without outliers, which is defined in [29] as:

$$\theta = \arccos|\hat{n}_1^T.\hat{n}_2|, \tag{19}$$

where $\hat{n}_1$ and $\hat{n}_2$ are the two unit normals from the fitted planes with and without outliers, respectively, (ii) the variation along the plane normal or the least eigenvalue λ_o, and (iii) the curvature $\sigma(p_i)$ as defined in Eq. (3).

5.1. Synthetic datasets

The synthetic datasets used in the following sections are generated by randomly drawing samples from two sets of multivariate 3D (x, y, z) Gaussian normal distributions, one set for regular observations and the other set for outlying cases. We create the Regular R observations assuming that they are from a planar surface, hence the variations among the points in the *z* or out-of-the-plane direction is significantly lower that the variations in the in-plane *x* and *y* directions. The regular observations in 3D have means of (2, 2, 2) and variances of (6, 6, 0.01). Usually, the Outlying O cases are far from the planar surface, so we create the outlying cases with means (7, 6, 8) and variances (2, 2, 1.5). We simulate the datasets for different sample sizes (*n*) and Outlier Percentages (OP) as needed.

5.1.1. Accuracy and robustness

To evaluate the accuracy of the plane parameters, we calculate the bias angle θ in Eq. (19). To get statistically significant results, we simulate 1000 sets of 50 3D points including 10 or 20% outliers which follow a Gaussian normal distribution with the same mean and variance parameters as described in Section 5.1. Fig. 3a depicts the pattern of a dataset of 50 points including 10 outlying cases, with regular points marked as black and the outliers marked as red stars that appear to be clustered. Planes fitted by the different methods are shown in Fig. 3b, in which PCA, qs_p and uLSIF planes of all the points is tilted away from the real plane of 40 regular points with a large θ, and the planes of all the points from the robust methods (RANSAC, MSAC, RPCA, MCMD_Z and MCMD_MD) are almost aligned with the plane without outliers. From the results from the 1000 runs, we calculate various descriptive measures including mean, median and Standard Deviation (StD) of θ as shown in Table 1. Results show that in every case of mean, median and StD the proposed methods have lower values than the others. PCA has the largest values for all the measures. Based on the values of average θ in Table 1 we can arrange the methods according to their rank of overall superiority in descending order as: MCMD_Z, MCMD_MD, RPCA, MSAC, RANSAC, uLSIF, qs_p and PCA. We use the well-known box plot as a robust visualisation tool, which gives insight into the descriptive measures of θ from the 1000 runs. In Fig. 4a, it is seen that the PCA and qs_p boxes stand out from the boxes of the other methods. Results from column 7 in Table 1 and the lengths of the boxplots support the fact that MCMD_Z and MCMD_MD have the 50% of θ values within the minimum Quartile Range (QR=3rd quartile -1st quartile) 0.295 and 0.402, respectively. That means the two proposed methods produce more robust results than the others. In Fig. 4b, we exclude the boxplots for PCA and qs_p so that the figure better presents the robustness for the robust and diagnostic statistical methods (RPCA, MCMD_Z and MCMD_MD) than RANSAC, MSAC and uLSIF. The '+' signs in the box plots show that the proposed methods have less outlying results. We see in Table 1 and Fig. 4, that RANSAC and MSAC perform almost equally. Since RANSAC is more popular in the literature, for the rest of the paper we do not consider MSAC.

To investigate the influence of uniform outliers on different methods for plane fitting, we simulate 1000 sets of 50 points including 10 or 20% outliers which follow an Uniform distribution within -9 to +9 for all three axes(x, y, z) and the regular observations are as in the previous experiment. Fig. 3d portrays a dataset with uniform outliers. Results in columns 8-13 of Table 1 show that uLSIF and qs_p has been improved significantly having values for θ of mean 2.55 and 4.61 degrees, respectively. That means qs_p and uLSIF perform well in the presence of uniform outliers. Box plots for the results are shown in Fig. 4d.

To visualize the performance for a high percentage of outlier contamination, we generate two datasets of 50 points contaminated with 70% cluster outliers and 80% uniformly scattered outliers, with the fitted planes presented in Figs. 3c and 3f, respectively. We see that only MCM_MD and uLSIF successfully fit the planes. The other methods cannot tolerate such a high percentages of outliers which is discussed in Section 5.1.2. Fig. 3f shows that in the presence of 80% scattered outliers, RANSAC fits the plane almost at the right orientation. However it is influenced by outliers and the size of the plane is enlarged. That means some outlying points work as inliers, which is the well-known masking effect. Figs. 3 (b, c, e and f) show that MCMD_MD is not affected by such a limitation. We carry out 1000 runs for the datasets of 50 points with 70% clustered outliers. Results are in the boxplots in Fig. 4c that shows that only MCMD_MD gives robust estimates in the presence of 70% outliers.

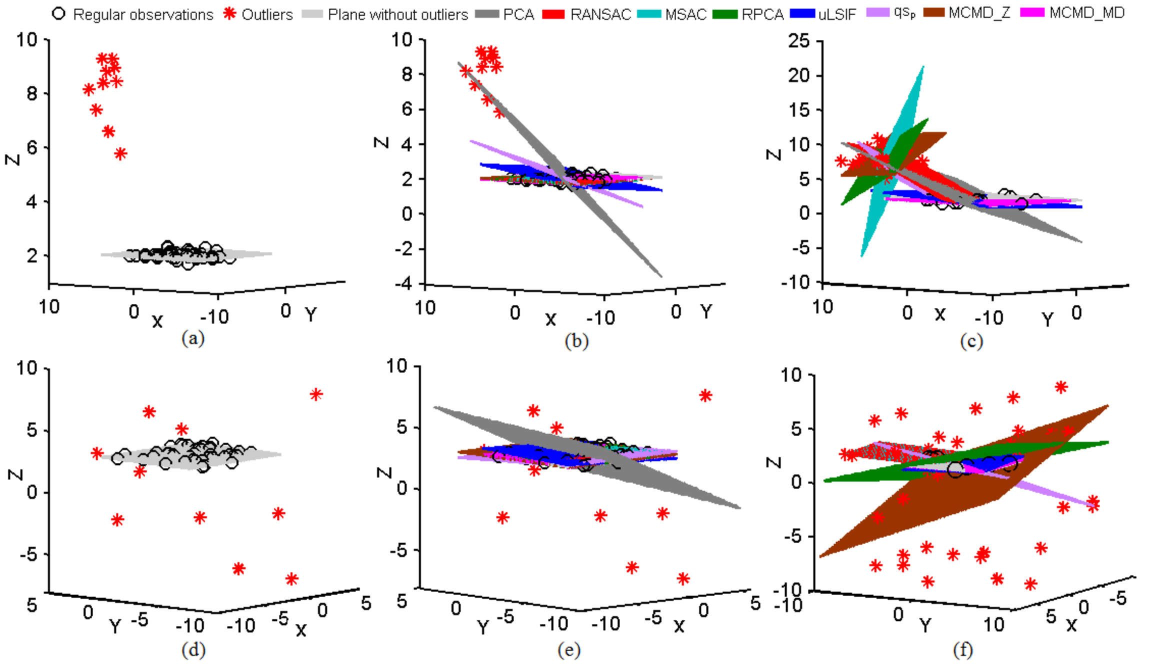


Fig. 3. (a) Dataset of 50 points with 20% clustered outliers, fitted planes: (b) n=50, OP=20, (c) n=50, OP=70; and (d) dataset of 50 points with 20% uniform outliers, fitted planes: (e) n=50, OP=20, (f) n=50, OP=80.

Table 1
Descriptive measures for bias angles (in degrees) from different methods.

Methods	Cluster outliers						Uniform outliers					
	Mean	Min.	Max.	Median	StD	QR	Mean	Min.	Max.	Median	StD	QR
PCA	34.483	3.807	52.690	34.255	3.973	4.758	27.593	0.442	89.819	25.339	16.231	19.237
RANSAC	1.188	0.000	6.367	0.832	1.167	1.618	1.184	0.005	9.140	0.824	1.204	1.574
MSAC	1.140	0.000	7.378	0.687	1.215	1.605	1.157	0.000	7.379	0.730	1.202	1.569
RPCA	0.694	0.022	2.698	0.599	0.489	0.550	0.675	0.025	3.326	0.550	0.504	0.547
uLSIF	6.097	0.562	17.938	5.731	2.304	2.769	2.550	0.056	12.277	2.116	1.874	2.146
qs_p	24.144	0.017	43.968	30.262	15.302	30.168	4.611	0.034	18.722	3.603	3.775	4.666
MCMD_Z	0.389	0.007	1.896	0.328	0.268	0.295	0.427	0.016	1.905	0.366	0.286	0.353
MCMD_MD	0.514	0.005	2.119	0.450	0.319	0.402	0.522	0.012	2.244	0.452	0.335	0.420

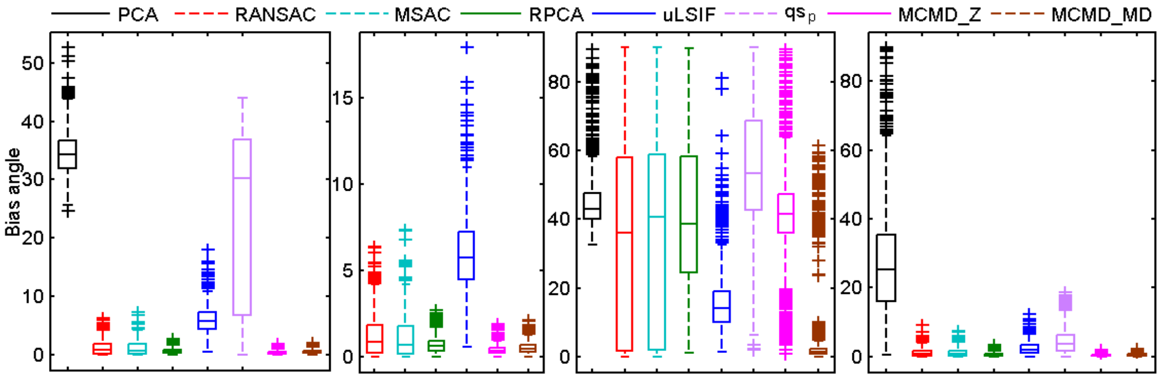


Fig. 4. Box plots of bias angles for n=50, OP=20, clustered outliers: (a) all the methods (b) all methods excluding PCA and qs_p, (c) all the methods n=50, OP=70, and (d) box plot for all methods n=50, OP=80, uniform outliers.

Results in Table 1 illustrates that the proposed methods significantly perform better with more robust results than other methods for clustered as well as uniform outliers. For brevity of space, in the next sections for simulated data, we evaluate the performances only in the presence of clustered outliers as shown in Fig. 3a.

5.1.2. Breakdown point evaluation

We use the bias angle θ to calculate the Breakdown Point (BP), which is used as a robustness measure [19]. A bias angle between the best-fit-planes from the data with and without outliers for a robust method should be virtually zero. We generate 1000 datasets of 100 points using the same parameters as for the previous experiment with clustered outlier percentages of 1 to 80. We calculate the values of θ from the fitted planes from different methods for every dataset. Fig. 5 show the results for average θ calculated from 1000 samples. Fig. 5a clearly shows that PCA breaks down in the presence of just one outlier. That means PCA has a BP = 0%. The values of average θ from PCA differ greatly from the zero line for every 1% to 80% of outlier contamination. Even for 1% of outliers uLSIF produces an average θ = 3.626°, and continues with an approximately linear pattern with between 1% to 80% outliers present. For PCA and qs_p, θ is increasing with the increase of outlier percentage, which indicates that the influence of outliers for those is unbounded. Fig. 5a shows that RPCA, MCMD_Z, RANSAC, and MCMD_MD break down approximately at 47%, 49%, 64% and 74% of outliers, respectively. The results show that MCMD_MD attains the highest BP. RPCA and MCMD_Z produce more accurate results with lessvalues of θ than RANSAC until they break down at 47% and 49% respectively. To explore the deviations of the methods, we exclude PCA and qs_p from Fig. 5b, and also uLSIF from Fig. 5c. Figures 5(a, b and c) clearly reveal the better performance of the proposed methods in the presence of outliers.

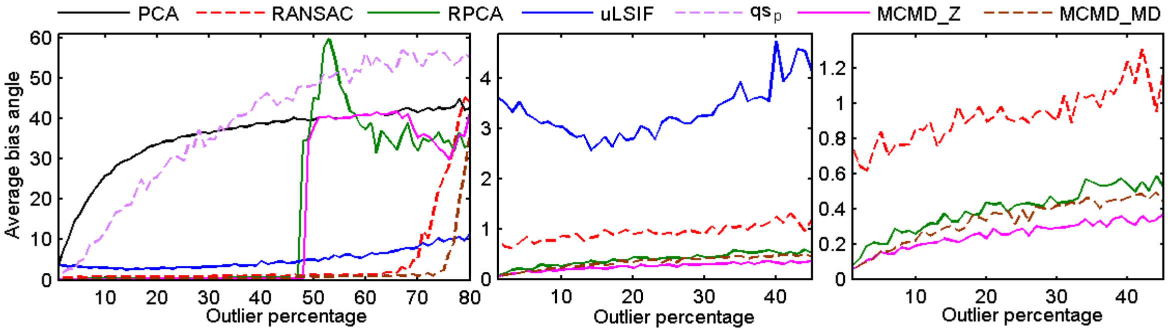


Fig. 5. Average θ versus outlier percentage, *n*=100: (a) all methods, (b) excluding PCA and qs_p, and (c) excluding PCA, qs_p and uLSIF.

5.1.3. Influence of sample size and outlier percentage on bias angles

To see the effect of sample size and different percentages of outlier presence in the data, we generate datasets for various sample sizes *n* of 20, 50 and 200, and outlier percentages 1% to 45%. We carried out 1000 runs for each and every sample size and outlier percentage. In the previous experiment, Fig. 5 shows that there are big gaps between non robust (PCA and qs_p) and the robust (RANSAC, RPCA, MCMD_Z and MCMD_MD) methods. Although uLSIF is not a robust method, we consider it with the robust methods because it produces significantly less values of θ. Results for average θ are shown in Fig. 6. In Fig. 6a, for a small sample *n*= 20, we see RANSAC gives inconsistent results for outlier percentages around 30% and more, and RPCA breaks down at 40% outliers. Results for *n* of 20, 50 and 200 (Figs. 6 a, b and c) show that uLSIF and RANSAC have larger values of θ than the other robust methods. MCMD_Z and MCMD_MD always perform better than RPCA, RANSAC and uLSIF. MCMD_Z has the least bias angles for almost all the cases of sample sizes and outlier percentages.

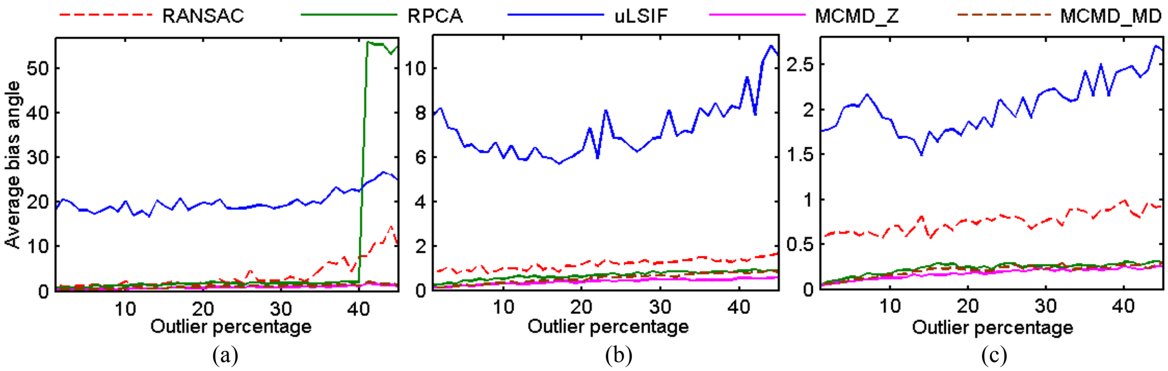


Fig. 6. Average θ versus outlier percentage for: (a) n=20, (b) n=50, and (c) n=200.

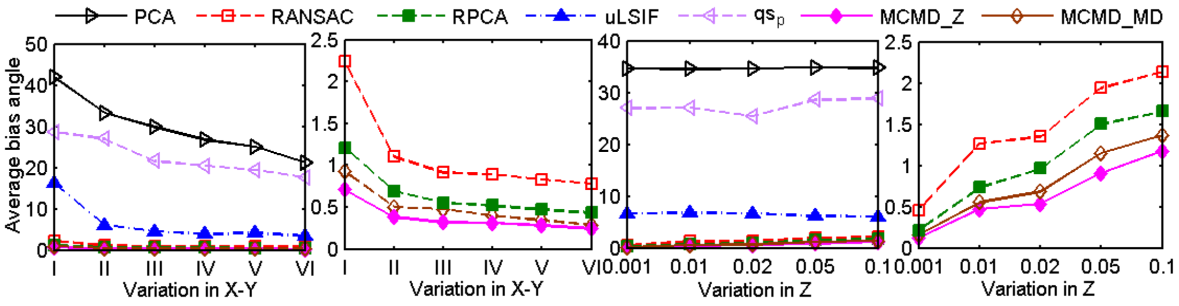
5.1.4. Effects of point density variation on bias angles

It is known that variations in point density affects plane parameter estimation and consequently θ. To see the effect, we simulate datasets with different variations in the two surface or x and y directions. We generate 1000 datasets of 50 points including 10 or 20% outliers for different combinations of variances in x and y. The rows of Table 2 show the combinations of variances for Regular R and Outlier O data. Other necessary parameters for the datasets are the same as used previously for clustered outliers. The results in Fig. 7a show that robust methods give low θ values. That is they are less influenced by outliers in the presence of point density variation, compared with PCA, qs_p and uLSIF. Fig. 7b shows that when we remove the results for PCA, qs_p and uLSIF, the proposed MCMD_Z and MCMD_MD values of θ are clearly lower than for RANSAC and RPCA.

Table 2
Variances for Regular R and Outlier O data.

Datasets	I	II	III	IV	V	VI
x(R,O) variances	(2,1)	(6,2)	(8,4)	(10,6)	(12,8)	(15,10)
y(R,O) variances	(2,1)	(6,2)	(8,4)	(10,6)	(12,8)	(15,10)

As well as point density variation, surface roughness influences surface fitting methods. To measure the effect of roughness on the estimates, we change the variance along the out-of-the-plane or z axis. We simulate 1000 datasets of 50 points with 20% outliers as for the previous experiments. The z variances for regular observations are 0.001, 0.01, 0.02, 0.05, and 0.1. Fig. 7c shows PCA, qs_p and uLSIF are markedly worse than the robust methods. Fig.7d excludes the results for PCA, qs_p and uLSIF, and shows that RANSAC has a larger and near steady increase in θ with respect to increase in z variance compared to the other robust methods. All the methods have increasing errors with the increase of *z* variance. MCMD_Z and MCMD_MD have improved accuracy than the other methods for all values of the *z* variance.



(a) (b) (c) (d)

Fig. 7. Line diagrams of average *θ* versus: (a) variances in *x*-*y* axes for all the methods, (b) variances in *x*-*y* axes for robust methods, (c) variances in z axis for all the methods, and (d) variances in *z* axis for robust methods.

5.1.5. Speed of processing

A major advantage of the proposed algorithms is speed of computation. We evaluate speed as a function of sample size and outlier percentage. We generate 1000 datasets of (i) different sample sizes 20, 50, 100, 1000, and 10,000 with a fixed percentage of 20% of outliers (shown in Table 3, columns 2 to 6), and (ii) different percentages of 5%, 10%, 20%, 40% and 50% outliers with a fixed 50 sample points (Table 3, columns 7 to 11). All the results for plane fitting in Table 3 are counted in seconds using the MATLAB® profile function. Results are average computation time from 1000 runs for each and every sample. The proposed methods always take significantly less time than RPCA, RANSAC and uLSIF. For example, for a sample size of 50 (Table 3, column 3), MCMD_Z and MCMD_MD take only 0.0085s and 0.0084s, respectively for fitting a plane. A simple calculation shows MCMD_Z (0.0085s) is approximately 5, 15 and 98 times quicker than uLSIF (0.0420s), RANSAC (0.1282s), and RPCA (0.8365s), respectively. These time gaps between existing (uLSIF, RANSAC and RPCA) and the proposed methods increase when the sample size increases to 10,000. However the times for the proposed methods increase at much lower rates than the times for uLSIF, RANSAC and RPCA. Time increases at a very high rate for uLSIF with the increase of sample size. For the different percentages of outliers, for example for 5% outliers, column 7 shows uLSIF (0.0430s), RANSAC (0.0734s) and RPCA (0.7937s) are approximately 9, 16 and 168 times slower, respectively than MCMD_Z (0.0047s). MCMD_Z takes slighly more time than MCMD_MD. Although PCA and qs_p take less time than the others, their accuracy is significantly worse (Table 1). Since after fitting a plane we get the saliency features, we can consider that the time for plane fitting would be same as the time for saliency feature estimation.

Table 3
Average computation time (in seconds) for different sample sizes and outlier percentages.

Methods	Sample size					Outlier Percentage				
	20	50	100	1000	10000	5	10	20	40	50
PCA	0.0004	0.0004	0.0005	0.0005	0.0016	0.0005	0.0004	0.0004	0.0005	0.0005
RANSAC	0.0714	0.1282	0.1845	0.4318	2.3934	0.0734	0.0934	0.1278	0.2562	0.3640
RPCA	0.8289	0.8365	0.8553	1.0751	1.2691	0.7937	0.7990	0.8007	0.8010	0.7937
uLSIF	0.0147	0.0420	0.1507	2.4642	139.7496	0.0430	0.0405	0.0440	0.0431	0.0409
qs_p	0.0009	0.0019	0.0037	0.0329	0.2910	0.0017	0.0019	0.0020	0.0019	0.0020
MCMD_Z	0.0088	0.0085	0.0098	0.0123	0.0400	0.0047	0.0064	0.0091	0.0238	0.0407
MCMD_MD	0.0079	0.0084	0.0092	0.0110	0.0376	0.0042	0.0054	0.0085	0.0228	0.0395

5.1.6. Outlier detection and performance as a classifier

In this section, we investigate the performance of all methods excluding PCA for outlier detection and as classifiers to categorize the points into inliers and outliers. We generate random datasets for sample size 100 with outlier percentages of 5%, 20% and 40% using the same input parameters used previously, for example as used for Fig. 3a.. We perform 1000 runs for each outlier percentage. We find and classify the points into inliers and outliers, and calculate the Correct Outlier Identification Rate (COIR), Correct Inlier Identification Rate (CIIR), number of inliers identified as outliers, and number of outliers identified as inliers, which are considered as True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR) or Swamping Rate (SR), and False Negative Rate (FNR) or Masking Rate (MR), respectively. We also compute accuracy based on true identification of outliers and inliers. The measures in percentages are defined in [54] as:

$$\text{TPR (COIR)} = \frac{Number\ of\ outliers\ correctly\ identified}{Total\ number\ of\ outliers} \times 100, \quad (20)$$

$$\text{TNR (CIIR)} = \frac{Number\ of\ inliers\ correctly\ identified}{Total\ number\ of\ inliers} \times 100, \quad (21)$$

$$\text{FPR (SR)} = \frac{Number\ of\ inliers\ identified\ as\ outliers}{Total\ number\ of\ inliers} \times 100, \quad (22)$$

$$\text{FNR (MR)} = \frac{Number\ of\ outliers\ identified\ as\ inliers}{Total\ number\ of\ outliers} \times 100, \quad (23)$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{Total\ number\ of\ points} \times 100, \quad (24)$$

where TP (True Positive) is the number of correctly identified outliers, and TN (True Negative) is the number of correctly identified inliers. Table 4 shows the average TPR, TNR, FPR, FNR and accuracy from 1000 runs.

Results show that RANSAC has the lowest rate of correctly identified inliers with a TNR of 33% and it is significantly affected by the swamping phenomenon as shown by a FPR of 67%. This means it misclassifies inliers as outliers at a very high rate. For uLSIF and qs_p with 20 % outliers (columns 9, 10), FNR is 6.2% and 51.2% , respectively. The methods are affected by the outliers and have larger values of θ (see Table 1) compared to the other robust methods. Our methods correctly identifies outliers with a very low FPR and without any FNs. For example, for the dataset with 20% of outliers; MCMD_Z and MCMD_MD have accuracies of 99.72% and 98.20% and SR of 0.4% and 2.3% respectively whereas RANSAC, RPCA, uLSIF, and qs_p have FPR of 66.2%, 3.7%, 5.3% and 25.3%, respectively. Hence, RANSAC fits a plane with a very high rate of misclassification error with lower support of inliers. For example, for 20% outliers, the RANSAC plane is fitted based on only 34 points (TNR of 33.8). Table 4 reveals: the proposed methods have the higher rate of TPR, TNR and accuracy with a very low rate of swamping and without masking for all the cases of outlier percentages.

To explore the variation in performance for classification (accurate inlier identification), we generate datasets of 100 points with 20% outliers and run the experiment 100 times. We calculate the number of inliers correctly identified and generate the histogram of the number of runs versus the number of inliers correctly identified for every run. In Fig. 8, the histograms show that most of the time the proposed methods perform better than the other methods and successfully identify inliers, whereas, RANSAC identifies inliers only around 20% to 40% out of the possible 80% inliers.

Table 4
Accuracy measures with classification into inliers and outliers with.

Methods	Outlier percentage														
	5					20					40				
	TPR	TNR	FPR	FNR	Accuracy	TPR	TNR	FPR	FNR	Accuracy	TPR	TNR	FPR	FNR	Accuracy
RANSAC	100.0	33.0	67.0	0.0	36.35	100.0	33.8	66.2	0.0	47.06	100.0	35.7	64.3	0.0	61.44
RPCA	100.0	94.3	5.7	0.0	94.62	100.0	96.3	3.7	0.0	97.05	100.0	98.3	1.7	0.0	98.97
uLSIF	76.8	96.6	3.3	23.2	95.68	93.9	94.7	5.3	6.2	94.54	96.9	87.9	12.1	3.1	91.50
qs_p	71.8	95.3	4.6	28.2	94.18	48.9	74.7	25.3	51.2	69.54	36.8	57.9	42.2	63.2	49.43
MCMD_Z	100.0	98.0	1.9	0.0	98.14	100.0	99.7	0.4	0.0	99.72	100.0	100.0	0.0	0.0	100.00
MCMD_MD	100.0	97.8	2.1	0.0	97.94	100.0	97.8	2.3	0.0	98.20	100.0	98.3	1.7	0.0	98.97

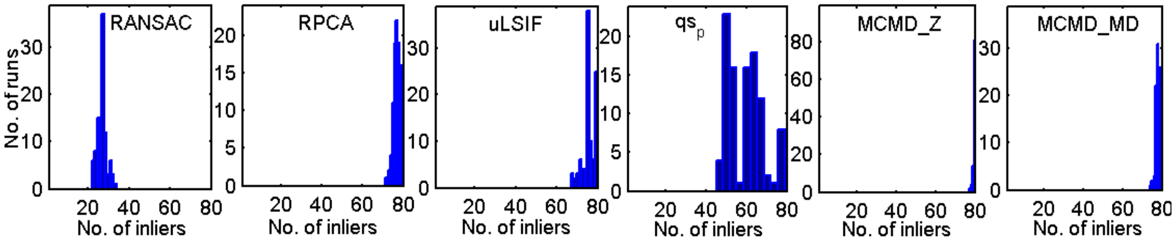


Fig. 8. Histograms for the number of runs versus the number of correctly identified inliers.

5.2. Real mobile laser scanning data

This section evaluates the performance of the proposed outlier detection methods for (i) outlier denoising, and estimating λ_o, curvature σ_p and the normal $\hat{n}$ for: (ii) sharp feature preserving, and (iii) segmentation in real Mobile Laser Scanning (MLS) data analysis.

5.2.1 Denoising in point cloud data

To demonstrate that the proposed algorithms are able to remove noise and to recover the detail from real point cloud data, we consider a vehicle-borne MLS dataset which is presented in Fig. 9a. The dataset contains 13,664 points of planar objects including a sign and the road surface, and non-planar objects including a pole and signposts. The data is very clean and contains a small amount of noise. However we deliberately add 10% (i.e. 1,366 points) Gaussian noise with mean 0 and StD 0.2 to determine how well the method deals with such high levels of noise. The noisy dataset is shown in Fig. 9b. To identify noise with the proposed algorithms we calculate the Rz score defined in Eq. (17) and used in Algorithm 2, and the RMD defined in Eq. (18) and used in Algorithm 3, for all the points based on their respective local neighbourhoods. The point p_i is defined as noise if Rz_i or RMD_i exceeds their respective cut-off values. We perform all the methods with neighbourhood size k of 50, and calculate the values of correct noise or outlier identification rate, inliers or real points identification rate, false positive rate and accuracy, which are similar to COIR (TPR) in Eq. (20) and CIIR (TNR) in Eq. (21), FPR in Eq. (22), and Accuracy in Eq. (24), respectively. We also count the number of Correctly Identified Noise/Outlier (CIN) and Correctly Identified Inlier (regular) points (CIR), FPR (swamping rates), and FNR (masking rates), respectively, with results given in Table 5. Results show that MCMD_Z has the highest accuracy of 93.57% with a minimum FPR of 6.58% and FNR of 4.90%. RPCA produces better results than RANSAC, uLSIF and qs_p but RPCA has higher rates of swamping of 12.08% and masking of 7.25% compared with the proposed methods. Although uLSIF and qs_p give better accuracies than RANSAC,both of them have higher values for FNR or masking. Figs. 9c and 9d are the results after removing the noise using MCMD_Z and MCMD_MD, respectively. These look similar to Fig. 9a before noise was added. Results also demonstrate that a few real noise points in the red circle in Fig. 9a (top left of the light) are removed in Fig. 9c and in Fig. 9d.

Table 5
Performance evaluation for outlier denoising.

Methods	CIN	CIR	TPR (%)	TNR (%)	FPR (%)	FNR (%)	Accuracy (%)
RANSAC	1086	10926	79.50	79.96	20.04	20.50	79.92
RPCA	1267	12014	92.75	87.92	12.08	7.25	88.36
uLSIF	658	12164	48.17	89.02	10.98	51.83	85.31
qs_p	641	12082	46.93	88.42	11.58	53.07	84.65
MCMD_Z	1299	12765	95.10	93.42	6.58	4.90	93.57
MCMD_MD	1282	12550	93.85	91.85	8.15	6.15	92.03

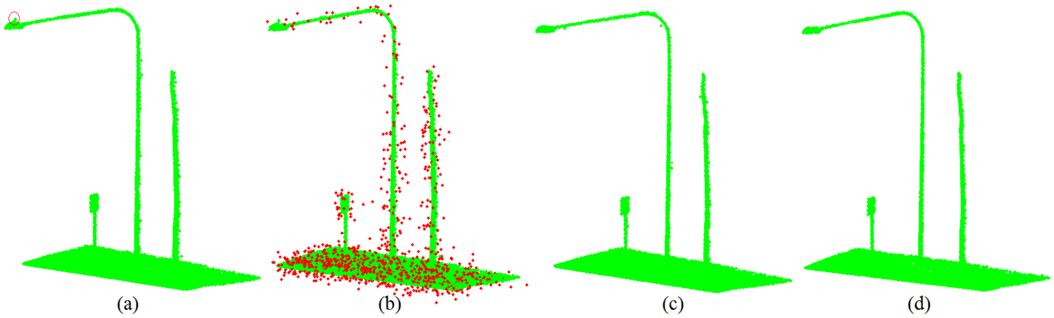


Fig. 9. (a) Real point cloud data, (b) point cloud data with 10% noise (red points), point cloud data after denoising: (c) results for MCMD_Z, and (d) MCMD_MD.

5.2.2. Sharp feature preserving

Many methods have been introduced for sharp features recovery [5, 6,16]. This task is not easy because normals on or near sharp features become overly smooth mainly for two reasons: (i) neighbourhood points come from multiple regions, and (ii) the presence of outliers and/or noise. The advantage of the proposed algorithms is they remove outliers and noise from the local neighbourhood and depend only on the majority of consistent observations. Hence they can automatically avoid the influence of outliers/noise and the points from other regions that are a minority in the local neighbourhood. The normal represents the best-fit-plane and the real surface from which the Maximally Consisten Set (MCS) comes from.

Regression-based techniques tend to smooth sharp features, and thus fail to correctly estimate normals near edges [32]. To illustrate this, we pick a part (Fig.10b) near an edge in a real MLS dataset (Fig. 10a), and estimate normals with neighbourhood k = 20. Fig. 10c shows that PCA fails to get perfect normals and makes the edge smooth. Results for uLSIF in Fig. 10f and for qs_p in Fig. 10g are similar to PCA. Fig. 10e show that the normal for RPCA appear to be more accurately classified than those for RANSAC shown in Fig. 10d. The results for MCMD_Z in Fig. 10h and for MCMD_MD in Fig. 10i show they efficiently construct the normals with correct orientation near edges and are hence able to retrieve sharp features better than the others.

For sharp feature recovery, we pick two small MLS datasets. Fig. 11a shows a box like object that contains 3,339 points and consists of edges and a corner. This we name the 'box like' dataset. Fig. 13a shows another dataset that is part of a crown shaped roof extracted from a roadside building. This we name the 'crown' dataset. The crown dataset is of 3,017 points and represents a polyhedral having bilinear surfaces with common edges and corners. We know that the angle of the tangent planes for bilinear surfaces varies along the edges and could cause problems for feature detecting and reconstructing systems using global sets of parameters [6].

We use a recently proposed algorithm [55] to extract the sharp features for the two datasets. The algorithm considers the i^{th} point as a sharp point on an edge or corner if its corresponding least eigenvalue is:

$$\lambda_o > \text{mean } (\lambda_o) + a\times\text{standard deviation } (\lambda_o) \ ; \qquad a = 1. \tag{25}$$

We fit planes for every point in the dataset with neighbourhood size k= 30, and calculate the least eigenvalues λ_o. Figs. 11b and 13b show that PCA is not good for recovering the edge or corner points correctly for both the datasets. RANSAC, uLSIF and qs_p do not successfully classify surface, edge or corner points. Many surface points appear as edge or corner points because of the smoothing effect around edges and corners. The results for MCMD_Z and MCMD_MD in Fig. 11g and Fig. 11h show that the new methods perform significantly more accurately than the others. In Figs. 13g and 13h, it can be seen that the proposed methods efficiently recover sharp features even in the presence of bilinear surfaces. To illustrate the performance of the methods in the presence of noise, we deliberately add 20% artificial Gaussian noise with mean 0.0 and StD. 0.1 to the dataset in Fig. 11a. The noisy boxlike dataset is in Fig. 12a. The results for PCA, RANSAC, uLSIF and qs_p show that in the presence of noise they are more greatly affected by noise than the earlier noise free results. More surface points are misclassified on and/or near edges and corners. The proposed methods still produce better results than the existing methods for edge and corner points as shown in Figs. 12g and 12h.

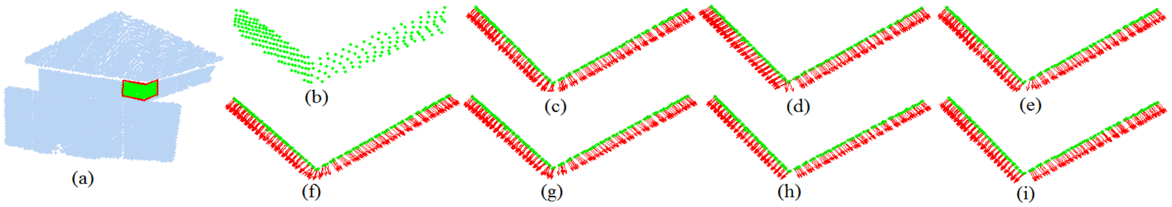


Fig.10. (a) Real point cloud data (b) sample data for normal estimation, normals plots: (c) PCA, (d) RANSAC, (e) RPCA, (f) uLSIF, (g) qs_p, (h) MCMD_Z, and (i) MCMD_MD.

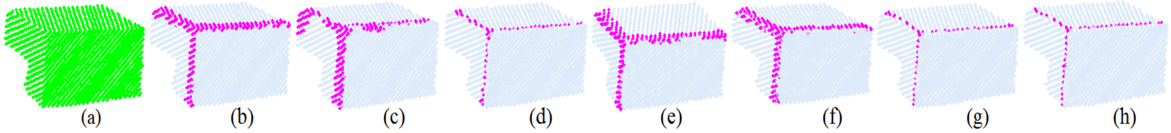


Fig. 11. (a) Boxlike dataset, edge and corner points recovery: (b) PCA, (c) RANSAC, (d) RPCA, (e) uLSIF, (f) qs_p, (g) MCMD_Z, and (h) MCMD_MD.

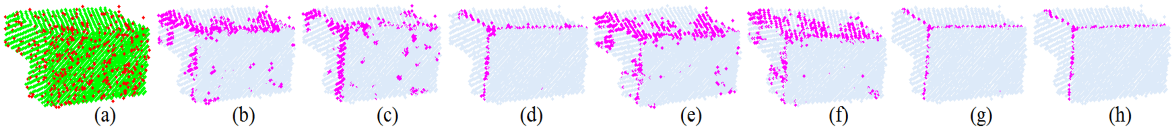


Fig. 12. (a) Boxlike dataset with noise, edge and corner points recovery: (b) PCA, (c) RANSAC, (d) RPCA, (e) uLSIF, (f) qs_p, (g) MCMD_Z, and (h) MCMD_MD.

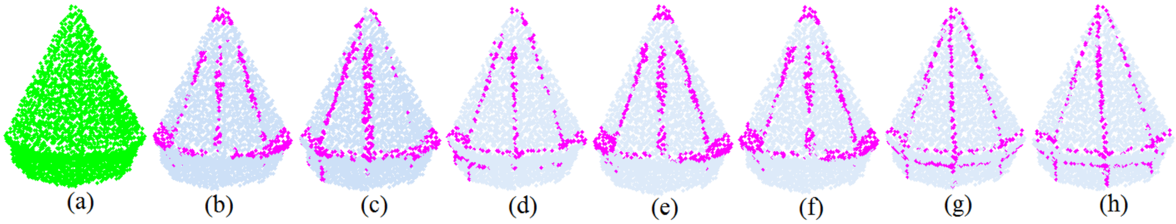


Fig. 13. (a) Crown dataset, edge and corner points recovery: (b) PCA, (c) RANSAC, (d) RPCA, (e) uLSIF, (f) qs_p, (g) MCMD_Z, and (h) MCMD_MD.

5.2.3. Segmentation

This section evaluates the estimated normals and curvatures from the proposed MCMD_Z and MCMD_MD based robust method realised in Algorithm 4, and the existing methods using a recently proposed segmentation algorithm [56]. Real MLS point cloud data is used and segmentation extracts homogenous points in the data and labels them as belonging to the same regions.

The segmentation algorithm is based on a region growing approach that starts by searching for a seed point, which has the minimum curvature value in the dataset. The algorithm grows regions using local surface point criteria, that is normal and curvature. The local surface point criteria for all the points in the data are calculated based on the k-nearest neighbourhood Np_i. The algorithm considers Euclidian Distance ED_{ij} between the seed point p_i and one of its neighbours p_j, Orthogonal Distance OD_j for the j^{th} point to the best-fit-plane of the i^{th} seed or interest point, and the angle difference θ_{ij} in Eq. (19) between the seed point p_i and p_j. A neighbour p_j of the seed point p_i will be added to the current region R_c and the current seed point list S_c if:

(i) $\mathrm{OD}_j < \mathrm{OD}_{th}$, (ii) $\mathrm{ED}_{ij} < \mathrm{ED}_{th}$, and (iii) $\theta_{ij} < \theta_{th}$,

where OD_{th}, ED_{th} and θ_{th} are the thresholds of the respective characteristics. OD_{th} and ED_{th} are fixed automatically within the segmentation algorithm, and θ_{th} is user defined. The region R_c will grow until no more candidate points are available, and a segment is considered as significant if its size is larger than a user defined threshold R_{min}. The reader is referred to [56] for more details about the segmentation algorithm.

Dataset 1 : Fig. 14a shows the first dataset acquired by a MLS system and consisting of 127,898 points. It describes parts of a road, kerb and footpath, and contains road side furniture including road signs, and long and approximately cylindrical surfaces (signs and light poles). These 21 objects can be classified as 10 planar and 11 non-planar surfaces. We label the dataset as the 'traffic furniture' dataset. For the segmentation algorithm, we set k=50 and θ_{th} =15°, and minimum region size R_{min}=10. The segmentation results are shown in Table 6 and Fig. 14.

To measure the accuracy of the segmentation results, we calculate the recall (r, surface segmentation rate), precision (p, correctness of the segmented surface) and F-score (F, overall accuracy) using the rules [54,57] :

$$\mathrm{r} = \frac{\mathrm{PS}}{\mathrm{PS+US}} \times 100, \tag{26}$$

$$\mathrm{p} = \frac{\mathrm{PS}}{\mathrm{PS+OS}} \times 100, \tag{27}$$

$$\mathrm{F} = 2 \times \frac{\mathrm{r \times p}}{\mathrm{r+p}}, \tag{28}$$

where PS=Proper Segmentation, US = Under Segmentation, and OS = Over Segmentation.

The segmentation results for PCA and qs_p are very poor with PS of 5 and 6 respectively. RANSAC has a better PS of 12 but this is combined with an OS of 4 and a US of 3. PCA, RANSAC, uLSIF and qs_p (Figs. 14b-f) all fail to segment the road, kerb and footpath, which are under segmented into one surface. RPCA performs better than PCA, RANSAC, uLSIF and qs_p. MCMD_Z and MCMD_MD (Figs. 14g-h) segment all

10 planar and 11 non-planar surfaces without any OS or US. The results in Table 6 show that RPCA has 97.44% accuracy (column F) value, whereas PCA, RANSAC, uLSIF and qs_p have only 43.48%, 77.42%, 75.86% and 60.00% accuracy respectively. The proposed methods have perfect or 100% accuracy for all r, p and F measurements.

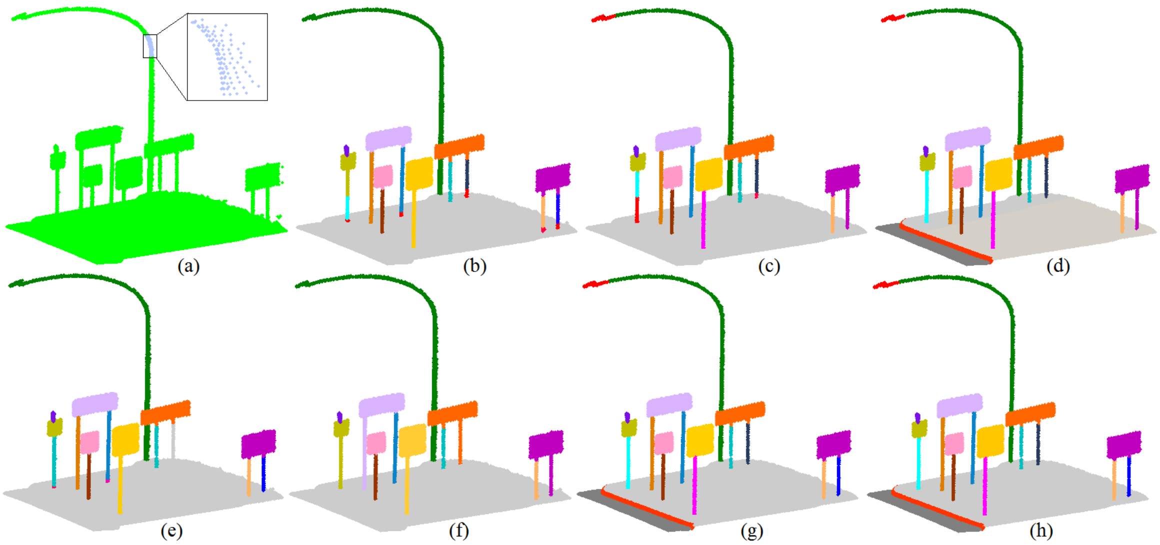


Fig. 14. (a) Traffic furniture data; segmentation results: (b) PCA, (c) RANSAC, (d) RPCA, (e) uLSIF, (f) qs_p, (g) MCMD_Z, and (h) MCMD_MD.

Table 6
Performance evaluation in segmentation.

Methods	Traffic furniture dataset							Bus stop dataset						
	TS	PS	OS	US	r (%)	p (%)	F (%)	TS	PS	OS	US	r (%)	p (%)	F (%)
PCA	22	5	7	6	45.45	41.67	43.48	50	12	27	5	70.59	30.77	42.86
RANSAC	20	12	4	3	80.00	75.00	77.42	42	20	16	2	90.91	55.56	68.97
RPCA	20	19	0	1	95.00	100.00	97.44	43	26	13	0	100.00	66.67	80.00
uLSIF	18	11	2	5	68.75	84.62	75.86	46	21	13	1	95.45	61.76	75.00
qs_p	13	6	0	8	42.86	100.00	60.00	44	20	15	2	90.91	57.14	70.18
MCMD_Z	21	21	0	0	100.00	100.00	100.00	37	29	5	0	100.00	85.29	92.06
MCMD_MD	21	21	0	0	100.00	100.00	100.00	40	29	4	0	100.00	87.88	93.55

Dataset 2: Fig. 15a shows the next dataset of 33,719 points also acquired by anMLS system. This dataset includes a bus shelter, bench, umbrella, light post, signs, road, kerb, footpath and a tree. We label this dataset as the ‘bus stop’ dataset. We use the same segmentation algorithm used for Traffic furniture dataset with the parameters k=50, θ_{th} =5° and R_{min}=10. It is clear from Fig. 15a that several surfaces are incomplete and the

point density is not homogenous. The dataset contains 31 different planar and non-planar surfaces. Fig. 15b shows that PCA fails to segment the road, kerb and footpath. uLSIF, qs_p, RANSAC and the other robust methods were able to segment the three surfaces namely the road, kerb and footpath properly. They also successfully preserve sharp features and segment the part of the roofs of the two umbrellas. Results for the bus stop dataset in Table 7 show that both of the proposed methods more accurately segment 29 surfaces out of 31 surfaces without any US. MCMD_Z and MCMD_MD have only 5 and 4 OS respectively, but for PCA the number of OS is 27 and for other methods RANSAC, RPCA, uLSIF and qs_p the OS is 13 or more. All the robust statistical methods have $\mathrm{r} = 100\%$ without any US. RPCA, uLSIF and qs_p have 80%, 75% and 70.18% accuracy respectively. The proposed MCMD_Z and MCMD_MD attain the accuracy (column F) of 92.06% and 93.55%, respectively.

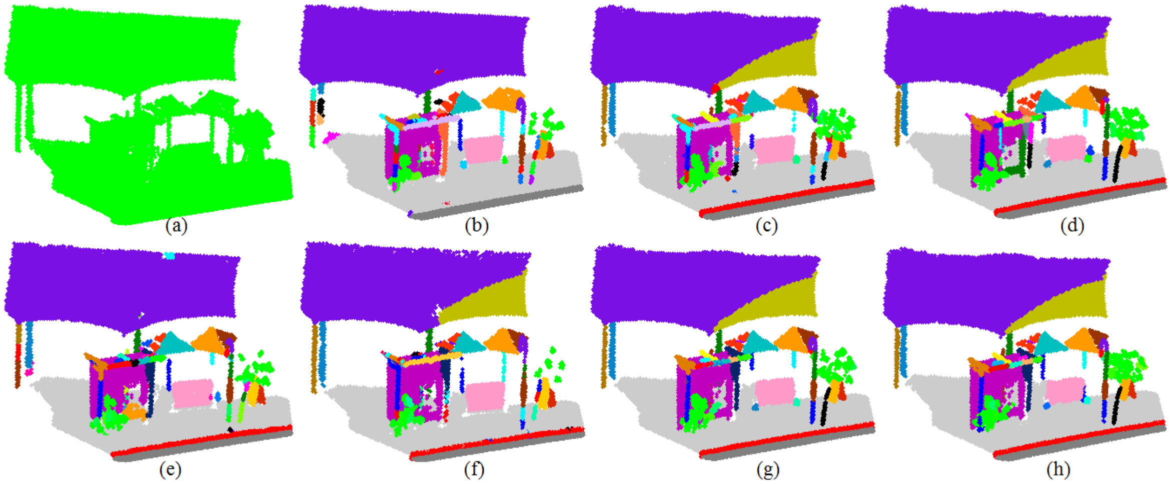


Fig. 15. (a) Bus stop data; segmentation results: (b) PCA, (c) RANSAC, (d) RPCA, (e) uLSIF, (f) qs_p, (g) MCMD_Z, and (h) MCMD_MD.

6. Conclusions

Using robust and diagnostic statistical approaches, two outlier detection and robust saliency feature estimation methods are proposed for mobile laser scanning 3D point cloud data analysis. In the proposed methods, basic ideas of robust and diagnostic statistics are coupled. First, the algorithms fit the best plane based on the majority of consistent data or inliers within the local neighbourhood of each point of interest. Then the outliers are found locally for every neighbourhood based on the results from the majority of good points. In the second stage, the required saliency features of normals and curvatures are estimated for every point by PCA based on the inlier points found in its local neighbourhood. Results show that PCA and qs_p are quicker than the proposed methods, but they are not as accurate. Although uLSIF produce significantly more accurate results than PCA and qs_p, all three are not robust. Results for artificial and real data show that the methods have various advantages over other techniques including: (i) being computationally simpler, (ii) being able to efficiently identify high percentages of clustered and uniform outliers, (iii) being more accurate than PCA, RANSAC, uLSIF and qs_p, (iv) being significantly faster than RPCA, uLSIF and RANSAC, (v) being able to denoise point cloud data, and (vi) being more efficient for sharp features recovery. In addition, the robust saliency features based on the proposed techniques can reduce over and under segmentation, and give significantly better segmentation results than existing methods for planar and non-planar complex surfaces.

ACKNOWLEDGMENTS

REFERENCES

[1] N. Amenta, Y. J. Kil, Defining point-set surfaces, ACM Transactions on Graphics, 23 (3) (2004) 264–270.

[2] T. K. Dey, L. Gang, J. Sun, Normal estimation for point cloud: a comparison study for a voronoi based method, in: M. Pauly, M. Zwicker (Eds.), Proceedings Eurographics Symposium on Point-Based Graphics, 2005, pp. 39–46.

[3] H. Hoppe, T. De Rose, T. Duchamp, Surface reconstruction from unorganized points, Proceedings of the ACM SIGGRAPA, vol. 26, no.2, 1992, pp.71–78.

[4] L. Klasing, D. Althoff, D. Wollherr, M. Buss, Comparison of surface normal estimation methods for range sensing applications, Proceedings of the IEEE International Conference on Robotics and Automation, 2009.

[5] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang, S. Jin, Robust normal estimation for point clouds with sharp features, Computers & Graphics 34 (2010) 94–106.

[6] C. Weber, S. Hahmann, H. Hagen, G-P. Bonneau, Sharp feature preserving MLS surface reconstruction based on local feature line approximations, Graphical Models 74 (6) (2012) 335–345.

[7] S. Sotoodeh, Outlier detection in laser scanner point clouds, in: International Archives of Photogrammetry, Remote Sensing and Spatial Information Systems, Dresden, vol. 36/5, 2006, pp. 297–302.

[8] M. Pauly, M. Gross, L. P. Kobbelt, Efficient simplification of point sample surface, Proceedings of the Conference on Visualization, Washington, D.C., 2002, pp. 163–170.

[9] T. Rabbani, Automatic Reconstruction of Industrial Installations Using Point Clouds and Images, PhD Thesis, NCG, Nederlandse Commissie voor Geodesie, Natherlands Geodetic Commission, Delft, The Natherlands, 2006.

[10] D. Belton, Classification and Segmentation of 3D Terrestrial Laser Scannar Point Cloud, PhD thesis, Department of Spatial Sciences, Curtin University of Technology, Australia, 2008.

[11] N. J. Mitra, A. Nguyen, L. Guibas, Estimating surface normals in noisy point cloud data, in: Spacial Issue of the International Journal of Computational Geometry and Applications 14 (2004) 261–276.

[12] A. Nurunnabi, D. Belton, G. West, Diagnostic-robust statistical analysis for local surface fitting in 3d point cloud data, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 1–3, 2012, pp. 269–274.

[13] E. Castillo, J. Liang, H. Zhao, Point cloud segmentation via constrained nonlinear least squares surface normal estimates, in: Michael Breuß, Alfred Bruckstein, Petros Maragos (Eds.) Innovations for Shape Analysis, Mathematics and Visualization, Springer-Verlag, Berlin Heidelberg, Chap. 13, 2013, pp. 283–299.

[14] V. Barnett, T. Lewis, Outlier in Statistical Data, John Wiley & Sons, New York, 1995.

[15] P. J. Rousseeuw, M. Hubert, Robust statistics for outlier detection, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1 (1) (2011) 73–79.

[16] S. Fleishman, D. Cohen-Or, C. Silva, Robust moving least-squares fitting with sharp features, ACM Transaction on Graphics 24 (3) (2005) 544–552.

[17] C. Aggarwal, Outlier Analysis, Springer, NY, USA, 2013.

[18] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review, 22 (2004), 85–126.

[19] P. J. Rousseeuw, A. Leroy, Robust Regression and Outlier Detection, John Wiley & Sons, New York, 2003.

[20] E. Schubert, A. Zimek, H.-P. Kriegel, Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection, Data Mining Knowledge Discovery 28 (2014) 190–237.

[21] I. T. Jollife, Principal Component Analysis, Springer, NY, USA, 1986.

[22] M. Hubert, P. J. Rousseeuw, K. V. Branden, ROBPCA: a new approach to robust principal component analysis, Technometrics 47 (1) (2005) 64–79.

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

[24] M. Zuliani, RANSAC for Dummies, http://vision.ece.ucsb.edu/~zuliani/Research/RANSAC/docs/RANSAC4Dummies.pdf, (2011).

[25] P. H. S. Torr, A. Zisserman, MLESAC: a new robust estimator with application to estimating image geometry, Journal of Computer Vision and Image Understanding 78(1) (2000) 138–156.

[26] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, T. Kanamori, Statistical outlier detection usisng direct density ratio estimation, Knowledge Information System 26 (2011) 309–336.

[27] T. kanamori, S. Hido, M. Sugiyama, A least-squares approach to direct importance estimation, Journal of Machine Learning Research, 10 (2009) 1391–1445.

[28] M. Sugiyama, K. M. Borgwardt, Rapid distance-based outlier detection via sampling, Proceedings of Advances in Neural Information Processing Systems (NIPS), Navada, USA, 5-9 December, 2013, pp. 467–475.

[29] C. Wang, H. Tanahashi, H. Hirayu, Comparison of local plane fitting methods for range data, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Kauai, vol. 1, 2001, pp. 663–669.

[30] J.-E. Deschaud, F. Goulette, A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing, Proceedings of the International Symposium on 3DPVT, Paris, 2010.

[31] H. Sheung, C. Wang, Robust mesh reconstruction from unoriented noisy points, SIAM/ACM Joint Conference on Geometric and Physical Modeling, New York, USA, 2009**,** pp.13–24.

[32] N. Amenta, M. Bern, Surface reconstruction by Voronoi filtering, Discrete and Computational Geometry, 22 (4) (1999) 481–504.

[33] A. Boulch, R. Marlet, Fast and robust normal estimation for point clouds with sharp features, in: Computer Graphics Forum, Blackwell Publishing Ltd, 31 (5) (2012) 1765–1774.

[34] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, Point set surfaces, Proceedings of the IEEE Visualization, 2001, pp. 21–28.

[35] A. C. Öztireli, G. Guennebaud, M. Gross, Feature preserving point set surfaces based on non-linear kernel regression, in: Computer Graphics Forum, Blackwell Publishing Ltd., vol. 28, no. 2, 2009, pp. 493–501.

[36] M. Breunig, H-P. Kriegel, R. T. Ng, J. Sander, LOF: Identifying density-based local outliers, Proceedings of the ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.

[37] E. M. Knorr, R. T. Ng, Algorithms for mining distance-based outliers in large datasets, Proceedings of the 24 International Conference on Very Large Databases (VLDB), NY, 1998, pp. 392–403.

[38] B. Scholkpf, J. C. Patt, J. Shawe-Taylor, A. J. Smola, Williamson, Estmating the support of a high-dimensional distribution, Neural Computation, 13 (7) (2001), 1443–1471.

[39] B. Liu, Y. Xiao, L. Cao, Z. Hao, F. Deng, SVDD-based outlier detection on uncertain data, Knowledge Information System 34 (2013) 597–618.

[40] H.-P. Kriegel, P. Kroger, E. Schubert, A. Zimek, LoOP: local outlier probabilties, Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM), Hongkong, 2009, pp. 1649–1652.

[41] S.Y. Jiang, Q.B. An, Clustering-based outlier detection method, Proceedings of the 5th IEEE International Conference on Fuzzy Systems and Knowledge Discovery, 2008, 429–433.

[42] D. Tax, R. Duin, Support vector data description, Machine Learning, 54(1) (2004), 45–66.

[43] A. Nurunnabi, A. S. Hadi, A. H. M. R. Imon, Procedures for the identification of multiple influential observations in linear regression, Journal of Applied Statistics, 41(6) (2014) 1315–1331.

[44] J. Feng, H. Xu, S. Yan, Robust PCA in high-dimension: A deterministic approach, Proceedings of 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.

[45] P. J. Rousseeuw, K. V. Driessen, A fast algorithm for the minimum covariance determinant estimator, Technometrics 41(3) (1999) 212–223.

[46] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, The Eurographics Association and Blackwell Publishing, 2007, pp. 1–12.

[47] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? Journal of the ACM 58 (3) (2011), Article No. 11.

[48] G. Li, Z. Chen, Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo, Journal of the American Statistical Association 80 (391) (1985) 759–766.

[49] H. Wang, D. Suter, Robust adaptive-scale parametric model estimation for computer vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1459–1474.

[50] B. J. Tordoff, D. W. Murray, Guided-MLESAC: faster image transform estimation by using matching priors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1523–1535.

[51] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers for Large data sets, Proceedings of the ACM SIGMOD, International Conference on Management of Data, 2000, pp. 427–438.

[52] M. Wu, C. Jermaine, Outlier detection by sampling with accuracy guarantees, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 767–772.

[53] H.-P. Kriegel, M. Scubert, A. Zimek, Angel-based outlier detection in high-dimensional data, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 442–452.

[54] T. Fawcett, Introduction to ROC analysis, Pattern Recognition Letters 27 (2006) 861–874.

[55] A. Nurunnabi, D. Belton, G. West, Robust segmentation for multiple planar surface extraction in laser scanning 3D point cloud data, Proceedings of the 21th International Conference on Pattern Recognition , Tsukuba Science City, Japan, 11–15 November 2012, pp. 1367–1370.

[56] A. Nurunnabi, D. Belton, G. West, Robust segmentation in laser scanning 3D point cloud data, Proceedings of the Digital Image Computing: Techniques and Applications (DICTA), Fremantle, Australia, 3–5 December 2012.

[57] W. E. Li, Q. Guo, M-K. Jakubowski, M. Kelly, A new method for segmenting individual trees from the Lidar point cloud, Photogrammetric Engineering and Remote Sensing 78(1) (2012) 75–84.