

A COMPARISON OF TWO NEW METHODS OF OUTLIER DETECTION FOR MOBILE TERRESTRIAL LIDAR DATA

M. Leslar^{a,*}, J.G. Wang^a, B. Hu^a

^a Dept. of Earth and Space Science and Engineering, York University, Keele Street, Toronto, Canada, M3J 1P3

Commission I, ICWG VI

KEY WORDS: Outlier Detection, Mobile Terrestrial LiDAR, curved surface fitting, Kalman Filter

ABSTRACT:

Terrestrial LiDAR provides many disciplines with an effective and efficient means of producing realistic three-dimensional models of real world objects. With the advent of mobile terrestrial LiDAR, this ability has been expanded to include the rapid collection of three-dimensional models of large urban scenes. For all its usefulness, it does have drawbacks. One of the major problems faced by the LiDAR industry today is the automatic removal of outlying data points from LiDAR point clouds. This paper will discuss the development and implementation of two methods of performing outlier detection in georeferenced point clouds. These methods will make use of the raw data available from most time-of-flight mobile terrestrial LiDAR scanners in both the temporal and spatial domains. The first method involves a moving fixed interval smoother derived from the well-known α - β - γ filter. The second method fits a quadratic curved surface to sections of LiDAR data. The use of these routines is discussed through examples with real LiDAR data.

1. INTRODUCTION

LiDAR (Light Detection and Ranging) is a tool, which allows for the fast and efficient capture of three-dimensional spatial information from real world targets. This ability has allowed both terrestrial based and airborne LiDAR to be used in a variety of applications (Antova, 2006; Lu, et al, 2003; etc.). Until recently, terrestrial based LiDAR has been relegated to stationary tripod mounts with comparatively low scanning speeds (2,500 points per second) when compared to airborne LiDAR systems (300,000 points per second). With the advent of mobile terrestrial LiDAR, this is no longer the case. Terrestrial based scenes can now be collected faster than ever, firstly because they are being collected from a moving platform and secondly, because collection speeds have greatly increased (200,000 points per second). This increase in the number of terrestrial based data points collected during a survey, means greater and greater amounts of data are being produced faster. To complicate matters, due to the fact that the scanners are now immersed in the scene being scanned, instead of flying high above it, the geometry contained in these massive data files is more complex than those encountered previously. This makes filtering of the data harder than previously encountered, but even more necessary. Specifically, detecting and eliminating erroneously collected points, or outliers, becomes critical.

Several sources (Sotoodeh, 2006; Lu, et al, 2003; Zheng, et al, 2008) provide definitions for the term outlier with LiDAR point clouds. Simply stated, an outlier is a point which differs from its neighbours or neighbourhood significantly. The determination of what the term significantly means is, of course, up to the individual user of the data. Outliers in LiDAR data occur for a variety of reasons. Some of these reasons, such as boundaries of occlusion, surface reflectance and multi-path

reflection are described in Sotoodeh (2006). To this list can be added moving objects which pass through the scan area faster than they can be captured and particulate matter, such as snow, rain, dust, etc., in the air, which reflect the laser energy.

In this study, two different algorithms for outlier detection are proposed and studied. These algorithms employ two separate concepts, one in the temporal domain and the other in the spatial domain. The algorithm in the temporal domain detects outliers by testing the difference between the raw measurement of a point and its predicted measurement as the central point of a specified data interval through a modified version of the moving fixed interval smoother derived from the α - β - γ filter (Wang, 1997). This method will make use of the precise timings available from LiDAR either mobile or static time-of-flight LiDAR in a PVA (Position, Velocity, Acceleration) Kalman filter to predict point positions based on its neighbourhood and remove those points which deviate from their predicted positions. The second algorithm involves using a best-fit quadratic curved-surface to spatially measure each point in the cloud and compare it to the points in its neighbourhood. Their performance is numerically studied. The sequential use of the spatial domain algorithm after the temporal domain algorithm should provide for an overall better result.

2. THE MATHEMATICAL MODELS

2.1 The Moving Fixed Interval Smoothing Method

Given a discrete time series $z(1), z(2), \dots, z(n)$ of a continuous time signal with their standard deviations $\sigma_1, \sigma_2, \dots, \sigma_n$ and the associated precise timings t_1, t_2, \dots, t_n , if $x(k)$ denotes the

* Corresponding author.

state of $z(k)$ at time instant t_k , the state equation of the α - β - γ filter is given by

$$x(k) = x(k-1) + \dot{x}(k-1)(t_k - t_{k-1}) + \frac{1}{2}\ddot{x}(k-1)(t_k - t_{k-1})^2 \quad (1)$$

$$\dot{x}(k) = \dot{x}(k-1) + \ddot{x}(k-1)(t_k - t_{k-1}) \quad (2)$$

$$\ddot{x}(k) = \ddot{x}(k-1) \quad (3)$$

without considering the process noise, and the measurement equation is

$$z(k) = x(k) + \Delta(k) \quad (4)$$

Where $\Delta(k)$ is the white noise with zero expectation and a variance of σ_k^2 . The moving fixed-interval smoother estimates the state for a time instant k (e.g. the central time instant) by using the measurements over a specified window (see Figure 1).

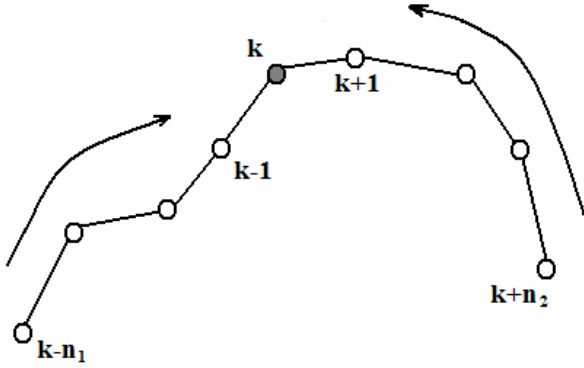


Figure 1: The Fixed Interval Smoother

With the measurements $z(k-n_1), \dots, z(k-1), z(k), \dots, z(k+n_2)$, the unbiased linear smoother can be derived based on the principle of minimal variance. The smoothed solution for the states at k is given by (Wang, 1997)

$$\hat{x}(k) = \sum_{i=-n_1}^{n_2} a_i(k) \cdot z(k+i) \quad (5)$$

$$\hat{\dot{x}}(k) = \sum_{i=-n_1}^{n_2} b_i(k) \cdot z(k+i) \quad (6)$$

$$\hat{\ddot{x}}(k) = \sum_{i=-n_1}^{n_2} c_i(k) \cdot z(k+i) \quad (7)$$

with their variances

$$\sigma_x^2(k) = \sum_{i=-n_1}^{n_2} a_i^2(k) \cdot \sigma_{k+i}^2 \quad (8)$$

$$\sigma_{\dot{x}}^2(k) = \sum_{i=-n_1}^{n_2} b_i^2(k) \cdot \sigma_{k+i}^2 \quad (9)$$

$$\sigma_{\ddot{x}}^2(k) = \sum_{i=-n_1}^{n_2} c_i^2(k) \cdot \sigma_{k+i}^2 \quad (10)$$

Where

$$a_i(k) = \frac{1}{\sigma_{k+i}^2} \{ \lambda_1(k) + \lambda_2(k) \cdot (t_{k+i} - t_k) + \lambda_3(k) \cdot (t_{k+i} - t_k)^2 \} \quad (11)$$

$$b_i(k) = \frac{1}{\sigma_{k+i}^2} \{ \mu_1(k) + \mu_2(k) \cdot (t_{k+i} - t_k) + \mu_3(k) \cdot (t_{k+i} - t_k)^2 \} \quad (12)$$

$$c_i(k) = \frac{1}{\sigma_{k+i}^2} \{ \eta_1(k) + \eta_2(k) \cdot (t_{k+i} - t_k) + \eta_3(k) \cdot (t_{k+i} - t_k)^2 \} \quad (13)$$

$$\begin{pmatrix} \lambda_1(k) & \mu_1(k) & \eta_1(k) \\ \lambda_2(k) & \mu_2(k) & \eta_2(k) \\ \lambda_3(k) & \mu_3(k) & \eta_3(k) \end{pmatrix} =$$

$$\left(\begin{array}{ccc} \sum_{i=-n_1}^{n_2} \frac{1}{\sigma_{k+i}^2} & \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}}{\sigma_{k+i}^2} & \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}^2}{\sigma_{k+i}^2} \\ \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}}{\sigma_{k+i}^2} & \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}^2}{\sigma_{k+i}^2} & \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}^3}{\sigma_{k+i}^2} \\ \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}^2}{\sigma_{k+i}^2} & \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}^3}{\sigma_{k+i}^2} & \sum_{i=-n_1}^{n_2} \frac{\delta_{k,k+i}^4}{\sigma_{k+i}^2} \end{array} \right)^{-1} \quad (14)$$

where $\delta_{k,k+i} = t_{k+i} - t_k$.

By rearranging equation (5), one can produce a model to predict the point $z(k)$:

$$z_p(k) = \frac{1}{1-a_0(k)} \left\{ \sum_{i=-n_1}^{-1} a_i(k) \cdot z(k+i) + \sum_{i=1}^{n_2} a_i(k) \cdot z(k+i) \right\} \quad (15)$$

The difference $\delta z(k) = z(k) - z_p(k)$ is used to detect outliers. The variance of this difference can be computed as follows:

$$\sigma_{\delta z}^2(k) = \sigma_k^2 + \frac{1}{(1-a_0(k))^2} \left\{ \sum_{i=-n_1}^{-1} a_i^2(k) \cdot \sigma_{k+i}^2 + \sum_{i=1}^{n_2} a_i^2(k) \cdot \sigma_{k+i}^2 \right\} \quad (16)$$

Accordingly, the standardized difference can be assumed to be normally distributed as

$$\frac{z(k) - z_p(k)}{\sigma_{\delta z}(k)} \quad (17)$$

under the Null hypothesis $H_0 : \delta z(k) = 0$ against the alternative hypothesis $H_1 : \delta z(k) \neq 0$.

A time series will be investigated from equation (15) where outliers will be identified through the use of an appropriate interval $(-n_1, n_2)$ as the predictor.

2.2 Quadratic Polynomial Surface Fitting

The generic model of a quadric curved-surface is given by

$$\begin{aligned} f(a_1, \dots, a_{10}, x, y, z) = \\ a_1 \cdot x^2 + a_2 \cdot y^2 + a_3 \cdot z^2 + a_4 \cdot x \cdot y + a_5 \cdot x \cdot z \\ + a_6 \cdot y \cdot z + a_7 \cdot x + a_8 \cdot y + a_9 \cdot z + a_{10} = 0 \end{aligned} \quad (18)$$

where (x, y, z) is the coordinate of a point on the surface and a_j ($j = 1, \dots, 10$) are the parameters. Due to the ambiguity in the surface determination introduced by the parameter a_{10} , it is necessary to constrain ten parameters by

$$\begin{aligned} C = a_1^2 + a_2^2 + a_3^2 + a_4^2 + a_5^2 + a_6^2 + a_7^2 + a_8^2 \\ + a_9^2 + a_{10}^2 = 1 \end{aligned} \quad (19)$$

Given the measurements (x_i, y_i, z_i) of point i with its 3x3 variance matrix D_{ii} and the approximate values $(a_1^{(0)}, \dots, a_{10}^{(0)})$ of ten parameters, one obtains the linearized form of (18) as

$$A_i v + B_i \delta a + w_i = 0 \quad (20)$$

where

$$v = \begin{pmatrix} \dots & \dots & \dots & v_{x_i} & v_{y_i} & v_{z_i} & \dots & \dots & \dots \end{pmatrix}^T \quad (21)$$

$$A_i = (a_{x_i} \quad a_{y_i} \quad a_{z_i}) \quad (22)$$

$$B_i = (x_i^2 \quad y_i^2 \quad z_i^2 \quad x_i y_i \quad x_i z_i \quad y_i z_i \quad x_i \quad y_i \quad z_i \quad 1) \quad (23)$$

$$w_i = F_i(a_1^{(0)}, \dots, a_{10}^{(0)}, x_i, y_i, z_i) \quad (24)$$

$$a_{x_i} = \frac{\partial F}{\partial x_i} = 2a_1^{(0)} x_i + a_4^{(0)} y_i + a_5^{(0)} z_i + a_7^{(0)} \quad (25)$$

$$a_{y_i} = \frac{\partial F}{\partial y_i} = 2a_2^{(0)} y_i + a_4^{(0)} x_i + a_6^{(0)} z_i + a_8^{(0)} \quad (26)$$

$$a_{z_i} = \frac{\partial F}{\partial z_i} = 2a_3^{(0)} z_i + a_5^{(0)} x_i + a_6^{(0)} y_i + a_9^{(0)} \quad (27)$$

Under the assumption that all of the measurement points are not correlated to each other, one can define

$$v_i = -A_i v = -\{a_{x_i} v_{x_i} + a_{y_i} v_{y_i} + a_{z_i} v_{z_i}\} \quad (28)$$

To create an equivalent single measurement to the measured three coordinate components of a point so that (28) can be simplified to

$$v_i = B_i \delta a + w_i \quad (29)$$

with

$$\sigma_i^2 = \begin{pmatrix} a_{x_i} & a_{y_i} & a_{z_i} \end{pmatrix} \cdot D_{ii} \cdot \begin{pmatrix} a_{x_i} \\ a_{y_i} \\ a_{z_i} \end{pmatrix} \quad (30)$$

for $i = 1, 2, \dots, n$.

As can be seen, the combination of (29) and the linearized form of (18) is a standard parametric adjustment model with constraint. There is no need to provide further detail for its solution. For more details, refer to (Wang, 2009).

The test statistic based on the standard residual of each of the equivalent measurements can generally be given by

$$T_i = \frac{|v_i|}{\hat{\sigma}_0 \cdot \sqrt{q_{v_i v_i}}} \sim t(n-9) \quad (31)$$

where $\hat{\sigma}_0$ is the posterior variance of unit weight and $q_{v_i v_i}$ is the cofactor of v_i .

Once a patch size is specified, one can attempt to spatially detect outliers by moving this curved-surface fitting process through the data set.

3. IMPLEMENTATION

The methods described above were implemented using Microsoft Visual C++ 6.0. Figure 2 shows the basic idea of how each of these algorithms works.

The moving fixed interval prediction algorithm recognizes the fact that the point cloud can be treated as a series of lines of point data (Figure 2(a)). Since a significant portion of any terrestrial LiDAR scan is likely to include portions of the sky, numerous LiDAR points can be expected to be missing from the point cloud. These missing shots effectively segment the continuous line being followed by the scanner's optics, into multiple smaller line segments. Treating these smaller line

segments as independent entities, allows us to apply the Kalman filter to each of these subset lines from the point cloud. Since only a section of the total point cloud is being used to compare the results from these three methods, allowances have to be made for lines shorter than the window size ($-n_1, n_2$) and the window size has to be adjusted to accommodate points at the start and end of each line.

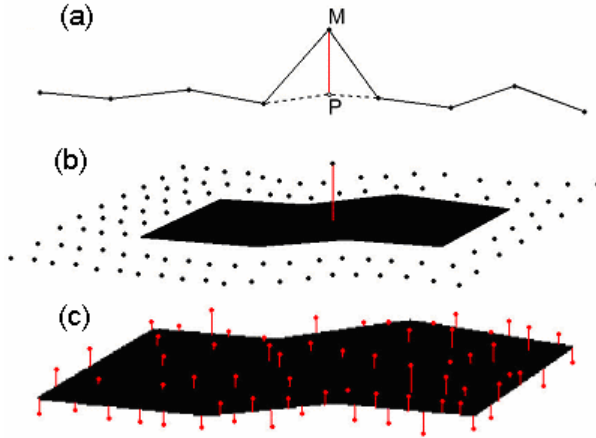


Figure 2: Methods for the Detection of Outliers. (a) Time series of points used to generate predictions (P) for measured points (M). (b) Polynomial surface patch in the immediate neighbourhood of the point being tested. (c) Spatial residuals to the best fit polynomial surface, used to test multiple points simultaneously.

The quadratic curved-surface fitting algorithm generates small surface patches in the neighbourhood of each point (Figure 2(b)). This is an outlier detector in the spatial domain, which relies on the assumption that the points immediately adjacent to an outlier will themselves lie on surface and not be outliers as well. The variable in implementing this method is the number of point cloud coordinates surrounding the point in question which should be used. On one hand, at least 10 points are required to derive the best fit surface, on the other hand, the larger the number of coordinates used, the greater the probability that other outliers will be incorporated into the calculation of the surface. In fact, when discussing LiDAR, the conditions, which cause an outlier, will also greatly increase the likelihood that other outliers lie close by. Therefore, care must be taken when setting a patch size. Instead of computing the polynomial surface for numerous small patches, this algorithm can also generate the curved surface for much larger sections of the point cloud. The idea is to segment the point cloud and compute the polynomial surface for user defined sections of the point cloud. Using the residual produced from the adjustment, one can examine the separation of each point from the surface. Using a test statistic, such as the one given in equation (31), outliers can be identified. Due to the potential discrepancies in the magnitude of any set of outliers, it might be necessary to discard identified outliers and re-compute the surface. Iteration in this manner should continue until no more outliers are

identified. This should ensure that all outliers are accounted for.

4. ALGORITHM TEST

The algorithms were tested using mobile terrestrial LiDAR data collected with the LYNX Mobile Mapper (www.optech.ca). Data was obtained on a section of asphalt from a generic parking lot, over which, multiple drive passes were performed. Figure 3 shows the three point clouds selected for testing and Table 1 gives specifics about the content of these point clouds.

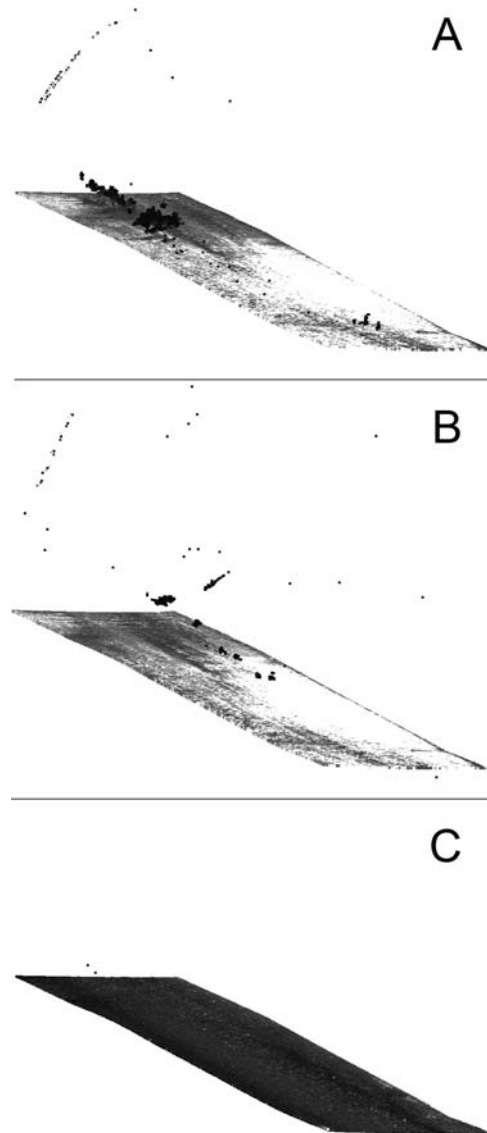


Figure 3: Point clouds of the same asphalt strip of parking lot, used during testing of the three outlier algorithms previously described. Point cloud A contains numerous outliers clustered together in two large groups above the asphalt surface. Point cloud B contains numerous outliers as well; however, these outliers are more evenly distributed above the asphalt surface. Point cloud C contains few outliers, most of which are within centimetres of the asphalt surface.

Table 1: Specifications for point clouds used in algorithm testing.

Point Cloud	A	B	C
Total No. of Points	1098689	295147	237740
Total No. of Outliers	11035	872	31
Total % of Points Which are Outliers	1.00	0.30	0.01

Point cloud A contains numerous outliers in two large groups. As shown in Table 1, the outliers make up 1.00% of the total point cloud. This data was collected on a day where the asphalt was wet, but the temperature was just below 0° Celsius. The prevailing cold wet conditions caused condensation from the vehicle’s exhaust pipe to combine with varying high and low intensity returns from the standing pools of water. This caused multiple laser reflections to be recorded above the asphalt surface.

Point cloud B was collected later the same day as point cloud A. There are far fewer outliers in this point cloud (0.30% from Table 1) and they are more spread out. Conditions were nearly the same, however the temperature had risen to just above 0° Celsius. A traffic barrier arm, which restricts vehicle access to the parking lot, caused the linear outliers in the top left hand corner of point clouds A and B. The LiDAR system captured data on this arm while the arm was in operation.

In contrast, point cloud C was collected on a day where the temperature was close to 20° Celsius and the pavement was dry. These conditions produced a point cloud with comparatively few outliers (0.01% from Table 1). Many of the outliers which do exist in this data set are within centimetres of the asphalt surface. The traffic barrier arm was not captured in operation in this scan.

The results from tests conducted using data strips A, B and C are given in Table 2 for algorithm (a), in Table 3 for algorithm (b) and in Table 4 for algorithm (c).

In addition, a combination of algorithms (a) and (b) was performed, where the reduced point cloud produced by algorithm (a) was input to algorithm (b). The results for this test conducted using data strips A, B and C are given in Table 5.

Table 2: Results from trials conducted using Algorithm (a) on point clouds A, B and C.

Point Cloud	A	B	C
No. of Outliers Identified	3414	291	4
No. of Non-Outliers Identified	55	27	3
No. of Outliers Missed	7621	581	27
% of Outliers Identified	30.94	33.37	12.90
% of Point Cloud Identified	0.32	0.11	0.00
% of Point Cloud Identified Incorrectly	0.01	0.01	0.00

Table 3: Results from trials conducted using Algorithm (b) on point clouds A, B and C.

Point Cloud	A	B	C
No. of Outliers Identified	16	33	4
No. of Non-Outliers Identified	0	0	0
No. of Outliers Missed	11024	839	27
% of Outliers Identified	0.14	3.78	12.90
% of Point Cloud Identified	0.00	0.01	0.00
% of Point Cloud Identified Incorrectly	0.00	0.00	0.00

Table 4: Results from trials conducted using Algorithm (c) on point clouds A, B and C.

Point Cloud	A	B	C
No. of Outliers Identified	5221	712	15
No. of Non-Outliers Identified	15431	0	193
No. of Outliers Missed	5060	160	16
% of Outliers Identified	47.31	81.65	48.39
% of Point Cloud Identified	1.88	0.24	0.09
% of Point Cloud Identified Incorrectly	1.40	0.00	0.08

Table 5: Results from trials conducted using Algorithm (a) preceding Algorithm (b) on point clouds A, B and C

Point Cloud	A	B	C
No. of Outliers Identified - Routine (a)	3449	290	4
No. of Outliers Identified - Routine (b)	11	24	1
No. of Non-Outliers Identified - Routine (a)	53	25	4
No. of Non-Outliers Identified - Routine (b)	1	0	0
No. of Outliers Missed	7575	558	27
% of Outliers Identified	31.35	36.01	16.13
% of Point Cloud Identified	0.32	0.11	0.00
% of Point Cloud Identified Incorrectly	0.00	0.01	0.00

5. DISCUSSION

Algorithm (a), was able to identify an average of 25% of the outliers between the three point clouds. The performance appears better on point clouds A and B, where there was a fairly well defined separation between the road surface and outlying data points. Most of the outliers in point cloud C lie just outside the standard deviation limit for an outlier and seem to be problematic for the routine. The non-outliers that were wrongly identified by the routine occur in areas of the point cloud where the regularly of the time series is disrupted by rough objects such as manholes. Where a manhole was encountered, the data density is insufficient to model the raised surfaces on the manhole’s lid. Also, areas of the point cloud where the system was collecting data while stationary,

occlusions and changes in surface direction seem to cause false detections. Despite these limitations the data indicate that the percentage of miss identified points for point clouds A and B are less than 10%.

Algorithm (b), performed worse at the removal of outliers than algorithm (a). The average percentage of outliers found and removed is only about 5%. This routine did exceptionally poorly on data where the outliers were clumped closely together as is the case in point cloud A. This routine had slightly more success on point cloud B, where the outliers are more spread out on the road surface. However, in areas of point cloud B where outliers are clumped together, this algorithm fails to distinguish the outlier from the surface. The performance of this algorithm was comparable with the previous one for point cloud C. In fact, since no points were selected in error, one could argue that the performance was slightly better.

Algorithm (c), removed the most outliers compared to the other methods. It also removed far more non-outlier data than the other routines. The results from point cloud B seem to indicate that this method works best with sparse outlier data that is significantly different from the surface. When the outlier data was clumped together (point cloud A) or the outlier data was close to the surface (point cloud C) the number of false alarms was excessive. Removal of so many non-outlier points limits the usefulness of this routine, however, it should be noted that in all cases the amount of the point cloud identified for removal is less than 1%.

Finally, the combinations of methods (a) and (b) was tried to see how much benefit could be gleaned. Do the high rate of false alarms; it was decided not to include method (c). The combination of the two routines produced results very similar to the addition of the results obtained when the two routines operated separately. This is as expected.

6. CONCLUSION

This paper has discussed two new algorithms for the detection and filtering of point clouds collected using mobile terrestrial LiDAR. These routines have taken advantage of the extra information that is generally available from this type of equipment. The two mathematical models presented here allowed for the creation of three computer routines which perform outlier detection in mobile terrestrial point cloud data.

The outlier detection methods, presented here, have proven to have value in the filtering and removal of outlier data from mobile terrestrial LiDAR point clouds. While each method has proven to have its own strengths and weaknesses, they have each proven capable of detecting and removing outliers from actual LiDAR data. More work is needed to optimize the inputs to the routines, specifically, to determine accurate error estimates for the point cloud coordinates.

7. REFERENCES

Antova, G., 2006, Precise Mapping with 3D Laser Scanning. Proceedings of the International Conference on Cartography and GIS, Borovets, Bulgaria, January 25-28, 2006.

Breuing, M.; H. Kriegel; R. Ng and J., Sander, 2000, LOF: Identifying Density-Based Local Outliers, Proc. of ACM

SIGMOD 2000 Int. Conf. On Management of Data, Dalles, TX, 2000.

Lu, C.T., Chen, D. and Kou, Y., 2003, Algorithms for Spatial Outlier Detection. Proc of the 3rd IEEE Int. Conf. on Data Mining (ICDM'03). November 19-22, 2003, Melbourne Florida, USA.

Miller, M.M., Meertens, C., Phillips, D., Rubin, C., Ely, L. and Pratt-Sitaul, B., 2009, Collaborative Research MRI: Acquisition of Terrestrial Laser Scanning Systems for Earth Science Research. Proposal Submitted to EAR Major Research Instrumentation, UNAVCO and Central Washington University.
http://www.unavco.org/pubs_reports/proposals/2009/MRI_TLS_EAR2009_UNAVCO-CWU.pdf (Accessed March 7, 2010).

Sotoodeh, S., 2006, Outlier Detection in Laser Scanner Point Clouds. Proceedings of the ISPRS Commission V Symposium, Image Engineering and Vision Metrology, Commission V, WG V/3, Volume XXXVI, Part 5, Dresden 25-27 September 2006. pp.297-302.

Wang, J.G., 2009, Least Squares Quadric Surface Fitting with the help of Statistical Tests– A Case Study in Industrial Surveying. International Geomatics Forum, Qingdao, China, May 29-30.

Wang, J.G., 1997, Pre-processing of INS–Data with the help of the α - β - γ -Filter, internal report, Insitute of Geodesy, UniBw Munich, July 1997. (in German)

Zheng, Min-qi; Chong-cheng, C.; Jia-xiang, L.; Ming-hui1, F. and Tamas, J., 2008, An Algorithm for Spatial Outlier Detection Based on Delaunay Triangulation. Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems (CISIS'08). October 23-24, 2008, Genova, Italy.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions of Optech Incorporated for their support in providing funding and the LiDAR data presented here. Without the data provided by Optech Incorporated, this research would not have been possible.