IMPROVING AND EXTENDING THE INFORMATION ON PRINCIPAL COMPONENT ANALYSIS FOR LOCAL NEIGHBORHOODS IN 3D POINT CLOUDS

David Belton

Cooperative Research Centre for Spatial Information (CRC-SI)
The Institute for Geoscience Research (TIGeR)
Department of Spatial Sciences, Curtin University of Technology
Perth WA, Australia -
d.belton@curtin.edu.au

Commission V, WG 3

KEY WORDS: Laser Scanning, Point clouds, Curvature Approximation, Surface Normal Estimation, Principal Component Analysis

ABSTRACT:

Principal Component Analysis (PCA) is often utilised in point cloud processing as provides an efficient method to approximate local point properties through the examination of the local neighbourhoods. This process does sometimes suffer from the assumption that the neighbourhood contains only a single surface, when it may contain multiple discrete surface entities, as well as relating the properties from PCA to real world attributes. This paper will present two methods. The first is a correction method to filter out the presence of multiple surfaces through an iterative process. The second is to combine the PCA preformed on the neighbourhood of point coordinates and normal approximations in order to estimate the radius of curvature in the maximum and minimum curvature directions.

1. INTORDUCTION

Point cloud processing relies on the analysis and examination of different observed attributes such as position, intensity and colour. Although these attributes are sampled directly from the laser scanner, attributes are more often derived from the examination of the local neighbourhood surrounding a point of interest. These estimated attributes, which include curvature, surface normal and geometric surface properties, are of great importance in point cloud-processing procedures such as for surface classification and segmentation.

Often these attributes are estimated with the use of Principal Component Analysis (PCA) performed on the local neighbourhoods of points, as it efficiently retrieves the local properties of a neighbourhood (Gumhold et al., 2001). Some common uses have included approximating the normal direction (Mitra et al., 2004), fitting first order planar surfaces (Weingarten et al., 2003), approximating surface curvature (Pauly et al., 2002), defining the tensors for tensor voting (Tong et al., 2004) and providing a local point coordinate systems (Daniels et al., 2007).

There are two problems that can occur when using PCA. The first is that a neighbourhood may contain multiple discrete surface entities. For most attributes, they are calculated under the assumption that there is only one surface structured. The affect of multiple surfaces can cause biases in the attributes e.g. the surface normal approximation. While deceasing the size of the neighbourhood may help in reducing the probability that a neighbourhood contains more than one sampled surface, the neighbourhood needs to be of sufficient size in order to reduce the effect of random errors and noise.

The other problem is how to relate the PCA results to the surface attributes. Often information can be lost, such as the approximation of curvature through surface variation (Pauly et al., 2002) where the comparable level of curvature is indicated, but there is no directional component or unit of measurement associated with the approximation.

It is the aim of this paper to present an iterative method of adjusting the neighbourhood to remove effects of multiple surface entities. In addition, a formula between the eigenvalues of the PCA and the surface properties such as the radius and direction of local curvature will be presented. From this formula, the maximum and minimum curvature directions can be also calculated in a closed form solution and this information provides a novel and detailed information of the neighbourhood of the point of interest.

2. PRINCIPAL COMPONENT ANALYSIS

PCA is performed by first calculating the covariance matrix Σ by the formula:

$$\Sigma = \frac{1}{k}\sum_{i=1}^{k}(x_i - c_0)(x_i - c_0)^T \qquad (1)$$

where x_i is defined as the vector form of the position of the i^{th} point in the neighbourhood containing the nearest k points and c_0 represents the centroid of the neighbourhood calculated as the mean of the neighbourhood. Since Σ is a symmetric and positive semi-definite matrix, it can be decompiled by eigenvalue decomposition such that the real positive eigenvalues, λ_0, λ_1 and λ_2, along with the corresponding

eigenvectors e_0, e_1 and e_2 form an orthogonal basis of the neighbourhood in $\mathbf{R}^3$ (Golub and Loan, 1989).

The covariance matrix Σ can be decomposed as follows:

$$\Sigma = \sum_{i=0}^{2} \lambda_i e_i e_i^T \tag{2}$$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$. Note that eigenvectors e_i represent the principal components, with the corresponding eigenvalues λ_i denoting the significance of each component in the form of the variance in these directions (Golub and Loan, 1989).

For a local neighbourhood of a point cloud, e_0 approximates the the local surface normal, with e_1 and e_2 approximating the tangential plane through c_0 (Pauly et al., 2002). This result is equivalent to a first order least squares plane fit (Shakarji, 1998).

3. IMPROVING NEIGHBOURHOOD SELECTION

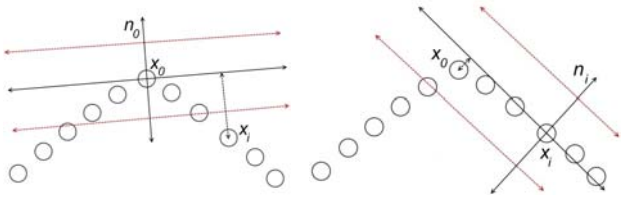
The initial neighbourhood can be selected in various ways. The problem is how to ensure that the assumption of the neighbourhood containing only one surface entity is valid. Techniques exist that remove multiple surfaces or reduce them down to a single dominate surface. Many are based on random sampling techniques, such as RANSAC (RANdom SAmpling Consensus) and its variants (Bolle and Vemuri, 1991), outlier detection (Danuser and Striker, 1998), voting methods (Page et al., 2002) and filtering (Tang et al., 2007).

This section aims to present a method that will iteratively converge to the correct solution for a neighborhood by adjusting the weights of points within a neighbourhood depending on the likelihood of that point being sampled from the dominate surface entity sampled within the neighbourhood. This will be done by examining the PCA of the neighbourhoods and, using statistical significance testing, determine and adjust weights iteratively for each point until the method converges to a stable solution (i.e. the weights remain constant). It will be demonstrated that the solution will have an uniform weighting for those points that are determined to belong to a dominate surface structure present in the neighbourhood, and zero for those that do not.

The first stage of the proposed method is outlining how to determine the relationship between two points within a neighbourhood and whether they belong to the same surface entity. This is separated into two models: an internal relationship between a point and the neighbourhood being corrected, and an external relationship between the point of interest and the neighbourhood of another point.

3.1 Internal and external neighbourhood relation between points

An internal relationship is defined as the relation of a point x_i to the neighbourhood N_0 surrounding a point of interest x_0 given that $x_i \in N_0$. Conversely, an external relationship is defined as the relation of a point of interest x_0 to the neighbourhood N_i surrounding x_i, given that $x_i \in N_0$. Illustration of the internal and external relationship is given in Figure 1(a) and Figure 1(b) respectively. An internal relationship of x_i to x_0 will be equivalent to an external relationship of x_0 to x_i.



(a) (b)

Figure 1: (a) Example of an internal relationship of x_i to x_0. The threshold is defined in red by Eq. 5 with all points inside considered to have an internal relationship. (b) Example of an external relationship of x_i to x_0. The threshold is defined in red by Eq. 6.

The main concept of this method is that a geometric attribute of a point x_0, which is related to its underlying surface, is dependent on its internal and external relationship with other surrounding points. In other words, the surrounding neighbourhood N_0 for the point x_0 should not only reflect the attributes within the neighbourhood, but x_0 should also reflect the attributes for the neighbourhood around x_i if they are to be considered to belong to the same surface entity.

To determine these relationships, two definitions of distance are utilised. Let c_0 and n_0 denote the centroid and normal approximation for neighbourhood N_0 around point x_0. The equation for the distance for the internal relationship is defined as:

$$dist_{ir} = \left| (c_0 - x_i) \bullet n_o \right| \tag{3}$$

with c_0 being calculated from the mean of the neighbourhood and n_0 is specified through PCA. In a similar manner, let c_i and n_i denote the centroid and normal approximation for neighbourhood N_i around point x_i. The equation for the distance for the external relationship is defined as:

$$dist_{er} = \left| (c_i - x_0) \bullet n_i \right| \tag{4}$$

Again, c_i is calculated from the mean of the neighbourhood and n_i is specified through PCA.

In order to determine if an internal or external relationship exists, the Boolean operators can be defined respectively as:

$$ir = \begin{cases} true & \rightarrow \quad dist_{ir} \leq t_{\upsilon, \frac{\alpha}{2}} s_0 \\ false & \rightarrow \quad dist_{ir} > t_{\upsilon, \frac{\alpha}{2}} s_0 \end{cases} \tag{5}$$

for the internal relationship and:

$$ir = \begin{cases} true & \rightarrow & dist_{ir} \le t_{\upsilon,\frac{\alpha}{2}s_0} \\ false & \rightarrow & dist_{ir} > t_{\upsilon,\frac{\alpha}{2}s_0} \end{cases} \quad (6)$$

for the external relationship. This Boolean operators come from a statistically significance test to determine if the relationship is likely to exist. s_0 and s_i are the error in the approximate normal direction for neighbourhood N_0 and N_i respectively. These values can be set as $\sqrt{\lambda_0}$ from the PCA of the respective neighbourhoods. The value t is the test statistic from the t-distribution with υ degrees of freedom and a significance factor of α.

3.2 Iterative generation of point membership weighting

PCA is performed to obtain the initial approximation for the variance and normal directions. The proposed method will use a slightly modified version of the covariance matrix formula such that:

$$\Sigma = \sum_{i=1}^{k} p_i (x_i - c_0)(x_i - c_0)^T \quad (7)$$

p_i is the weight of point x_i in the neighbourhood. Initially, the points will all be equally weighted as $p_i = 1/k$, with k being the number of points in the neighbourhood. In a similar manner, the formula for the centroid value will be modified to:

$$c_0 = \sum_{i=1}^{k} p_i x_i \quad (8)$$

If a point p_i is not related either internally or externally to point p_0, it is likely that it is not sampled from the same surface and the weight is decreased. Conversely, if a point p_i has an internal and external relationship to point p_0, then it is likely that they belong to the same surface, and the weight is increased. If there is only one relationship, then it is likely that at least one of the neighbourhoods (N_0 or N_i) is affected by multiply surfaces, and as such the weighs are left as it is until the neighbourhoods become more refined. From this, the rules for updating the weights are specified as:

$$p'_i = \begin{cases} p_i + \delta & \rightarrow & (\text{ir} = true) \text{and} (er = true) \\ p_i - \delta & \rightarrow & (\text{ir} = false) \text{and} (er = false) \\ p_i & \rightarrow & otherwise \end{cases} \quad (9)$$

where p'_i are the adjusted weights and δ is the small change to modify it by. If p'_i becomes a negative, then it is set to be zero in order to ensure all weights are non-negative. The new weights are then normalised by:

$$p''_i = p'_i \Big/ \sum_{i=1}^{k} p'_i \quad (10)$$

so that the summation of the weights equal unity. p''_i is then used as the weight in the recalculation of the covariance matrix for the next iteration.

3.3 2D case example

In this section, a 2D example of an intersection will be presented. The example is presented in Figure 2. As can be seen, the surface normals for the points near the edges are initially perturbed away from the normal direction of the surface they are sampled from as the neighbourhood is affected by more than one discrete surface structure. As defined method is applied, the weights are iteratively updated until they become stable, i.e. that successive iterations do not effect the solution and $p''_i \approx p_i$. The results are shown in Figure 3.

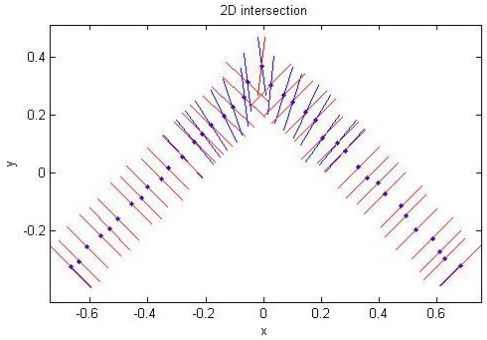


Figure 2: Example of correction applied to a 2D intersection. The blue lines represent the initial normal approximation and red lines represent the corrected normal approximations.

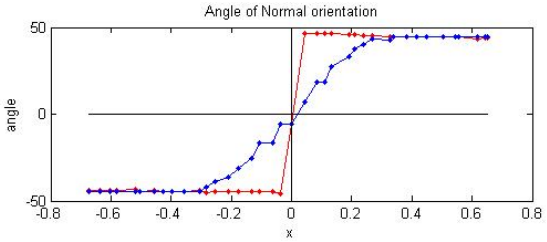


Figure 3: Angle of the normal orientation. The values for the surfaces should be approximately -45 and 45 degrees. The blue lines represent the orientation of the initial normal approximation and red lines represent the corrected values.

If the internal relationships were solely used to update the weights, the points whose neighbourhoods are only mildly affected by another surface structure are corrected as it behaves similar to an outlier removal process. If only the external relationships are used, while the results are similar to when both are used, the process can become unstable with all the points in the neighbourhoods being removed. If the weights of the points are examined at every iteration, as shown in Figure 4, it can be seen that they stabilised with either a zero value or an uniform value for those that are non-zero.

It should be noted that if the point closest to the intersection did not get corrected in order to conform to one of the surfaces. This is because the surfaces are equally represented in the neighbourhood and therefore one cannot be considered better than the other. The normal can be forced to align to one of the

surfaces by replacing the centroid c_0 with x_0. This instability caused in the covariance matrix allows it to align to one of the surfaces; however it can also affect the normal approximation detrimentally and provide biased solutions.

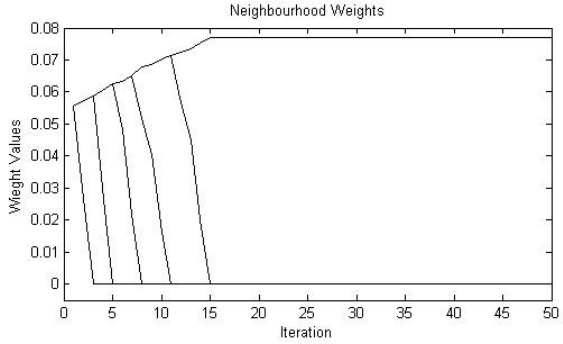


Figure 4: Trend of the weights for points in a neighbourhood as the iterations progress. The neighbourhood is affected by the presence of multiple surfaces with the points belonging to the dominant surface tending to a value of 0.077, and the others tending to a value of zero.

3.4 Practical Example

This section will demonstrate the outlined correction procedure as applied to a 3D point cloud. The point cloud, presented in Figure 5, is scanned from a door arch with a Leica ScanStation with a nominal point spacing of 0.01m. The correction procedure is applied to the point cloud with a neighbourhood size of 30 and the threshold for determining internal and external relationships set at a significance level of $\alpha = 0.5$.

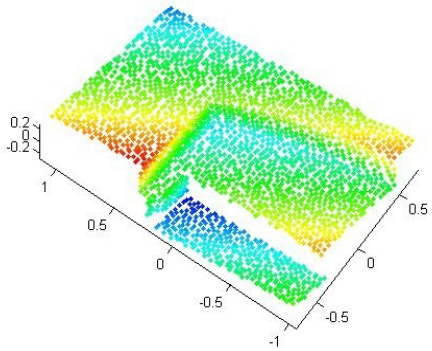
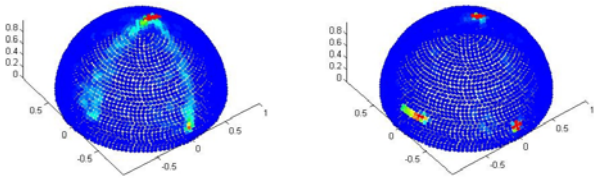


Figure 5: Point cloud sampled from a section of a door archway with a Leica Scanstation.

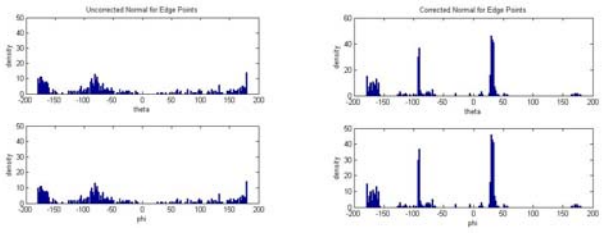
In Figure 6(a), the initial normal approximations are display on a Gaussian sphere. The clusters on the sphere represent the presence of surfaces with that normal orientation. The striping effect occurring between the clusters represents normal approximations being effected by more than one surface. Figure 6(b) shows the Gaussian sphere of the corrected normal approximations using the proposed method. As can be seen, the stripping effect is significantly reduced as those points affected by more than one surface are corrected to align with the dominant surface element in the neighbourhood. The neighbourhood weights are stabilised within 50 iterations of the procedure.



(a) (b)

Figure 6: (a) shows the Gaussian sphere of the uncorrected normal directions and (b) the Gaussian sphere of the corrected normal directions. The colour indicates the density of normal directions from blue representing zero to red representing in excess of a hundred.

On examining those points near a surface intersection Figure 8(a) shows the orientation angles for the uncorrected approximations and Figure 8(b) for the corrected approximations. This illustrates an increase in the accuracy of normal alignment after the correction with approximately 90% of the edge points now being aligned to within 5 degrees of their correct orientation.



(a) (b)

Figure 7: Histograms for edge points of the orientation angles for the normal directions. (a) is the uncorrected normal approximations and (b) is the corrected normal approximations. Peaks in the histogram denote orientation of the surface present in the point cloud.

4. APPROXIMATING CURVATURE

There are a variety of techniques to approximate curvature such as surface fitting (Besl and Jain, 1988), variation in surface normals (Jiang et al., 2005), tensor voting (Tong et al., 2004), and angles between neighbourhood members (Dyn et al., 2001). Two of the methods that involve PCA are given in Pauly et al. (2002) and Jiang et al. (2005). Pauly et al. (2002) uses the percentage of total population variation in the normal to determine the surface variation as a measure of curvature for a local neighbourhood. However, there is no unit or direction given to this curvature approximation. Jiang et al. (2005) provides an approximation of curvature based on the variation of the approximate normal directions in a local neighbourhood. While this measure as a directional component, it does not have a unit of measurement.

The advantages of both are their robust nature and fast computational time. A method will now be presented to combine the PCA of the coordinates and the normal directions for a neighbourhood to determine the principal directions of maximum and minimum curvature, in addition to the radius of curvature approximation in these directions.

4.1 Principal component analysis on point coordinates

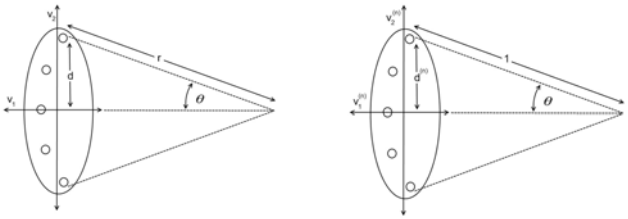
The properties that can be derived from PCA have been described previously. In the case of curvature, the value of λ_0 describes that amount of surface variation in the normal direction. Since curvature can cause a variation in the normal direction, λ_0 can be used to measure the level of curvature. In most cases λ_0 is divided by the total variation to provide an approximation (Pauly et al., 2002), otherwise the measure will be dependant on the span of the neighbourhood (Belton and Lichti, 2006). It is the span of the neighbourhood that is of interest in this method, and is described by λ_1 and λ_2. In the conic section displayed in Figure 8(a), if the distance d and the angle θ is known, then it becomes a simple matter of determining the radius of curvature by the following equation:

$$\sin\theta = d/r \tag{11}$$

and d can be approximated statistically using confidence interval as:

$$d = s\sqrt{\lambda_1} \tag{12}$$

with λ_1 being the standard deviation in this direction calculated from PCA of the neighbourhood, and s being a scale factor or number of standard deviations on either side of the mean that covers the span of the points. This leaves the only unknown in the formula as θ.



(a) (b)

Figure 8: (a) Conic section for the neighbourhood of point coordinates. (b) Conic section for the neighbourhood of point normal directions.

4.2 Principal component analysis on normal estimation

As illustrated in Jiang et al. (2005), the PCA performed on the normal directions of a neighbourhood can also provide an approximation of curvature. In this case, the covariance matrix Σ is defined as:

$$\Sigma = \frac{1}{k}\sum_{i=1}^{k}\left(n_i - \mu^{(n)}\right)\left(n_i - \mu^{(n)}\right)^T \tag{13}$$

with n_i denoting the normal direction and $\mu^{(n)}$ denoting the mean of the normal directions. The results from the eigenvalue decomposition with the eigenvalues $\lambda_0^{(n)}$, $\lambda_1^{(n)}$ and $\lambda_2^{(n)}$, along with the corresponding eigenvectors $e_0^{(n)}$, $e_1^{(n)}$ and $e_2^{(n)}$ such that $\lambda_0^{(n)} \le \lambda_1^{(n)} \le \lambda_2^{(n)}$. In this case, $\lambda_2^{(n)}$ represents the variation of the normal direction along the direction of maximum curvature (denoted by $e_2^{(n)}$) and $\lambda_1^{(n)}$ represents the variation of the normal direction along the direction of minimum curvature (denoted by $e_1^{(n)}$).

The conic section for the normals, displayed in Figure 8(b) can be examined in a similar manner to the conic section for the point coordinates. In this case, θ is solved as:

$$\sin\theta = d^{(n)}/1 \tag{14}$$

with $d^{(n)}$ being approximated as:

$$d^{(n)} = t\sqrt{\lambda_1^{(n)}} \tag{15}$$

with $\lambda_1^{(n)}$ being the standard deviation in this direction calculated from PCA of the normal, and t being a the number of standard deviations that covers the span of the normal variation.

4.3 Radius of curvature approximation

If the value of θ is the same in Figure 8(a) and Figure 8(b), then it becomes a simple matter of solving the radius of curvature. Assuming that there is only a single curved surface, by examining Figure 9, it can be seen that the neighbourhood for the normal directions is a scaled down version of the neighbourhood of point coordinates by a factor of r. The reason for this is that the scaling of the coordinate values would occur along the normal direction for each point. Therefore, the information for the PCA on the neighbourhood of point normals can be used to solve for the value of θ.

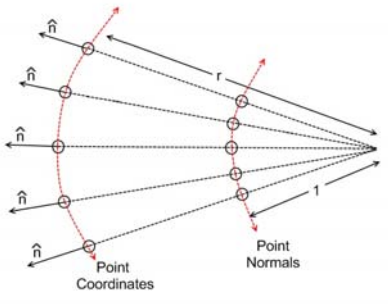


Figure 9: Neighbourhood of point normals overlaid on the neighbourhood of point coordinates. Depicts how neighbourhood of coordinates is scaled only along the normal direction of each point to get the neighbourhood of the normals.

Combining Eq. 11 and Eq. 14 and rearranging, the resulting equation for the radius of curvature is:

$$r = \frac{s\sqrt{\lambda_1}}{t\sqrt{\lambda_1^{(n)}}} \tag{16}$$

If it is assumed that they follow the same distribution (which is valid since one is a scaled version of the other), the values for *s* and *t* will be equal, thus cancelling out to leave the approximate radius of curvature as follows:

$$r = \sqrt{\lambda_1 / \lambda_1^{(n)}} \tag{17}$$

For the 2D case, this is simple as the principal components will be nominally aligned between the PCA on the neighbourhood of coordinates and normal. In the 3D case, they are often not aligned. Therefore, in order to use the approximation, the variance in the neighbourhood of point coordinates must be determined for the direction specified by $e_1^{(n)}$ and $e_2^{(n)}$, the two principle curvature directions. This can be done by using the ellipsoid defined by the eigenvector decomposition (Golub and Loan, 1989) to determine the variance in these directions. If λ'_1 and λ'_2 denote the variance of neighbourhood of point coordinates in the directions of $e_1^{(n)}$ and $e_2^{(n)}$ respectively, then radius of curvature in the minimum curvature and maximum curvature direction can now be defined respectively as:

$$r_{\min} = \sqrt{\lambda'_1 / \lambda_1^{(n)}} \tag{18}$$

$$r_{\max} = \sqrt{\lambda'_2 / \lambda_2^{(n)}} \tag{19}$$

4.4 Practical Application

This section will present the approximate radius of curvature measure applied to a practical data set. The point cloud used is shown in Figure 10 and was captured with a Leica 4500.

The results are shown in Table 1. *r* is the radius of curvature obtained through the pipe fitting routine in Cyclone, μ_r is the mean value for the approximated radius of curvature on for the pipe section and σ_r is the standard deviation of the approximated values. While simulated experiments perform well, the noise in a practical data set causes errors and variations in the results. The results may be improved by applying a smoothing filter beforehand.

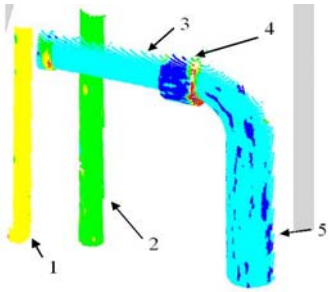


Figure 10: Point cloud sampled from an industrial scene containing multiple pipe sections. Colours are done based on a crude cluster analysis to identify different pipe radii.

Pipe Label	*r*	μ_r	σ_r
1	0.215m	0.240m	0.0147m
2	0.310m	0.240m	0.0298m
3	0.588m	0.600m	0.0020m
4	0.820m	0.823m	0.0751m
5	0.630m	0.646m	0.0837m

Table 1: Results for the approximated radius of curvature.

5. CONCLUSION AND FUTURE WORK

PCA is an important tool for deriving the local point properties through the examination of a local neighbourhood. In this paper, two methods were presented to utilise the information gained from PCA. The first method was an iterative correction procedure applied to a neighbourhood to filter out points deemed likely to belong to a different surface. The second was a method to combine the PCA performed on the point coordinates with the PCA performed on the point normal approximations to approximate the maximum and minimum curvature directions and the radius of curvature in each direction.

While these results are still preliminary, there is an indication that these methods could be used to retrieve information for procedures such as classification and segmentation. Future work will focus on examining the error propagation of point noise to increase the accuracy of the information.

ACKNOWLEDGEMENTS

This work has been supported by Curtin University of Technology, the Cooperative Research Centre for Spatial Information, who activities are funded by the Australian Commonwealth's Cooperative Research Centres Programme, and The Institute for Geoscience Research (TIGeR).

REFERENCES

Belton, D. and Lichti, D. D., 2006. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI(part 5), pp. 44–49.

Besl, P. J. and Jain, R. C., 1988. Segmentation through variable-order surface fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(2), pp. 167–192.

Bolle, R. M. and Vemuri, B. C., 1991. On three-dimensional surface reconstruction methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(1), pp. 1–13.

Daniels, J. D., Ha, L., Ochotta, T. and Silva, C. T., 2007. Robust smooth feature extraction from point clouds. In: IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07), Lyon, France, pp. 123–136.

Danuser, C. and Striker, M., 1998. Parametric model fitting: From inlier characterization to outlier detection. IEEE Transaction on Pattern Anaylsis and MAchine Intelligence 20(2), pp. 263–280.

Dyn, N., Hormann, K., Kim, S. J. and Levin, D., 2001. Optimizing 3D triangulations using discrete curvature analysis. In: Proceedings of Mathematical methods for curves and surfaces (Oslo 2000), pp. 135-146.

Golub, G. H. and Loan, C. F. V., 1989. Matrix Computations, 2nd edn, Johns Hopkins Press, Baltimore, MD.

Gumhold, S., Wang, X. and MacLeod, R., 2001. Feature extraction from point clouds. In: 10th International Meshing Roundtable.

Jiang, J., Zhang, Z. and Ming, Y., 2005. Data segmentation for geometric feautre extraction from lidar point clouds. Geoscience and Remote Sensing Symposium, 2005. IGARSS '05, pp. 3277–3280.

Mitra, M. J., Nguyen, A. and Guibas, L., 2004. Estimating surface normals in noisy point cloud data. International Journal of Computational Geometry and Applications 14(4,5), pp. 261–276.

Page, D. L., Sun, Y., Koschan, A. F., Paik, J. and Abidi, M. A., 2002. Normal vector voting: Crease detection and curvature estimation on large, noisy meshes. Journal of Graphical Models 64(3/4), pp. 199–229.

Pauly, M., Gross, M. and Kobbelt, L. P., 2002. Efficient simplification of point-sampled surfaces. In: VIS '02: Proceedings of the conference on Visualization '02, IEEE Computer Society, Boston, Massachusetts, pp. 163–170.

Shakarji, C. M., 1998. Least-squares fitting algorithms of the NIST algorithm testing system. Journal of Research of the National Institute of Standards and Technology 103(6), pp. 633–641.

Tang, Q., Sang, N. and Zhang, T., 2007. Extraction of salient contours from cluttered scenes. Pattern Recognition 40(11), pp. 3100–3109.

Tong,W.-S., Tang, C.-K., Mordohai, P. and Medioni, G., 2004. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3D. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), pp. 294–611.

Weingarten, J., Gruener, G. and Siegwart, R., 2003. A fast and robust 3d feature extraction algorithm for structured environment reconstruction. In: Proceedings of 11th International Conference on Advanced Robotics (ICAR), Portugal.