Automated Registration Of Unorganised Point Clouds From Terrestrial Laser Scanners

Article · July 2004

Source: OAI

CITATIONS
94

READS
222

2 authors, including:

Kwang-Ho Bae
FMG
16 PUBLICATIONS **553** CITATIONS

SEE PROFILE

AUTOMATED REGISTRATION OF UNORGANISED POINT CLOUDS FROM TERRESTRIAL LASER SCANNERS

Kwang-Ho Bae and Derek D. Lichti

Department of Spatial Sciences, Curtin University of Technology, GPO Box U1987, Perth, WA, 6845, Australia
baek@vesta.curtin.edu.au, d.lichti@curtin.edu.au

Commission V, WG V/2

KEY WORDS: Registration, Automation, Three-dimensional, Point Cloud, Laser Scanning, TLS

ABSTRACT

Terrestrial laser scanners provide a three-dimensional sampled representation of the surfaces of objects resulting in a very large number of points. The spatial resolution of the data is much higher than that of conventional surveying methods. Since laser scanners have a limited field of view, in order to obtain a complete representation of an object it is necessary to collect data from several different locations that must be transformed into a common coordinate system. Existing registration methods, such as the Iterative Closest Point (ICP) or Chen and Medioni's method, work well if good a priori alignment is provided. However, in the case of the registration of partially overlapping and unorganised point clouds without good initial alignment, these methods are not appropriate since it become very difficult to find correspondence. A method based on geometric primitives and neighbourhood search is proposed. The change of geometric curvature and approximate normal vector of the surface formed by a point and its neighbourhood are used to determine the possible correspondence of point clouds. Our method is tested with a simulated point cloud with various levels of noise and two real point clouds.

1 INTRODUCTION

The registration of point clouds is one of the important steps in processing data from laser scanners. Excellent reviews about the existing methods can be found in Haralick et al. (1989) and Rusinkiewicz and Levoy (2001). To the best of the authors' knowledge, a method for the registration of partially overlapping point clouds from terrestrial laser scanners without good a priori alignment has not developed yet. The most significant reason is that it is very difficult to recover the correspondence between point clouds without good a priori alignment. In this paper a method for the registration of two partially overlapping point clouds measured from different locations using geometric primitives and neighbourhood search without good a priori alignment is proposed. Sharp et al. (2002) proposed a method based on Euclidean invariant features: curvature, second order moments, and spherical harmonics. Our method is based on the change of curvature rather than curvature and is a thresholding method such as Zhang (1994).

The Iterative Closest Point (ICP) algorithm is a reliable and popular method for point cloud registration (Horn, 1987, Besl and McKay, 1992). Horn developed an adjustment method for recovery of unknown transformation parameters with knowledge of the correspondence of point clouds. Besl and McKay developed the ICP using Horn's algorithm in conjunction with a neighbourhood search algorithm. If approximate a priori information about the point-to-point correspondence of two point clouds is provided, then the ICP can iteratively recover the rigidbody transformation that aligns two point clouds. It converges monotonically to a local minimum, which may or may not be the global minimum, and all points in a point cloud are assumed to have correspondence in the other cloud.

Suppose that there are two point clouds, C^1 and C^2, and they are measured from different locations. A point cloud C^1 has a set of N_{C^1} points, $\{p_1^1, \cdots, p_{N_{C^1}}^1\}$. Bold and normal letters represent a vector and a scalar, respectively. Let $||\mathbf{p}_i^1 - \mathbf{p}_j^2||$ be the distance between point p_i^1 of point cloud C^1 and p_j^2 of C^2. Let $CP(p_i^1, C^2)$ be the corresponding point in C^2 of a point p_i^1. The ICP algorithm can be briefly described as follows.

1. Assume that the point in C^2 closest to a point in C^1 is the corresponding point.
2. Find the correspondence of two point clouds, $\mathtt{C} = \cup_{i=1}^{N_{C^1}} \{T_{iter=k}(p_i^1), CP(T_{iter=k}(p_i^1), C^2)\}$, where $\mathtt{C}$ is the set of all pairs of corresponding points, $T_{iter=k}$ is the transformation of the kth iteration and $T_{iter=0}$ is an initial transformation. $\mathtt{C}$ may or may not be one-to-one matching.
3. Compute the new transformation $T_{iter=k+1}$ that minimizes the sum of square distances between corresponding point pairs, i.e.

$$\sum_{i=1}^{n_{iter=k}} ||\mathbf{p}_i^1 - \boldsymbol{CP}(T_{iter=k}(p_i^1), C^2)||^2$$

   where $n_{iter=k}$ is the number of sample in kth iteration.

Since the ICP is based only on a local search algorithm to recover correspondence between two point clouds and it minimises the sum of square distances between possible corresponding points, it converges slowly sometimes and tends to fall into local minima.

Another algorithm is Chen and Medioni's that is a point-to-surface algorithm whereas the ICP is a point-to-point algorithm (Chen and Medioni, 1992). It minimises the sum of the square distances of a point to its corresponding surface. This algorithm is generally faster than the ICP. However, the point clouds need to be more closely aligned to each other initially than with the ICP.

The ICP, its variants, and Chen and Medioni's algorithm assume that the closest point in point cloud C^2 is a good estimate of the correct corresponding point in C^1. If two point clouds are not approximately aligned using a priori georeferencing information, this assumption is not correct. Although initial alignment can be provided from other means like surveying of the laser scanner locations, it is not always possible. Finding corresponding points and good registration of the point clouds are more difficult if they only partially overlap. In addition, these adjustment algorithms provides a closed-form solution, i.e. no iteration, which is one of reasons for their popularity, although closed-form methods can not provide statistical information of individual parameter of rigidbody transformation as conventional least square methods do.

2 PROPOSED METHOD

Geometric primitives, such as surface normal vector, curvature, and the change of curvature and so on, may provide additional and useful information to recover the correspondence of two point clouds. A method to find the correspondence of two point clouds using geometric primitives and a local search algorithm is proposed. Geometric curvature and the change of curvature is invariant to three dimensional rigid motion and surface normal vector can be rotated according to the computed transformation by Horn's or Chen and Medioni's algorithm. The angle between normal vectors and the difference between the changes of curvature of a point and its corresponding points are our criteria for selection of corresponding point pair.

The angle between approximate normal vectors of p_i^1 and p_j^2 can be expressed as

$$\theta_{(p_i^1;p_j^2)} = cos^{-1}(\mathbf{n}_{p_i^1} \cdot \mathbf{n}_{p_j^2}), \tag{1}$$

where $\mathbf{n}_{p_i^1}$ and $\mathbf{n}_{p_j^2}$ are the respective approximate normal vectors of the points. The difference in changes of curvature between two points can be written

$$\beta(p_i^1, p_j^2) = |M_{cc}(p_i^1) - M_{cc}(p_j^2)|, \tag{2}$$

where $M_{cc}(p_i^1)$ and $M_{cc}(p_j^2)$ are the approximate changes of curvature of p_i^1 and p_j^2. The normal vector of a point is estimated by covariance analysis of the point and its neighbourhood points and the change of curvature is estimated as the ratio of eigenvalues of the covariance matrix.

2.1 Normal vector estimation

The normal vector of a point is estimated by one of the eigenvectors of the covariance matrix of a point and its neighbourhood. The covariance of a point and its k neighbour points, $COV(p_i^1)$, is expressed as

$$COV(p_i^1) = \frac{1}{k}(\mathbf{p}_i^1 - \mathbf{p}^{cent}_{neighbour\{j=1\cdots k,p_i^1\}})^T (\mathbf{p}_i^1 - \mathbf{p}^{cent}_{neighbour\{j=1\cdots k,p_i^1\}}) \tag{3}$$

where $\mathbf{p}^{cent}_{neighbour\{j=1\cdots k,p_i^1\}}$ is the cetroid of p_i^1 and its k neighbourhood points. $COV(p_i^1)$ is a 3×3, real, positive, and semi-definite matrix, the eigenvalues of which are always greater than or equal to zero. The eigenvector of the minimum eigenvalue is the approximate normal vector of the surface formed by p_i^1 and its k neighbourhood points (Hoppe et al., 1992). The other eigenvectors are the tangential vectors of the surface. If the minimum eigenvalue is close to zero, then the surface consisting of a point and its neighbourhood is flat. This method is the first order approximation of the normal vectors of the surface. If the level of noise is large or the number of points in the neighbourhood, k, is too small, it could provide an incorrect normal vector.

2.2 Change of geometric curvature estimation

The change of geometric curvature at a point can be estimated from the eigenvalues of the covariance matrix. Each eigenvalue represents the spatial variations along the direction of the eigenvector. Let λ_i and $\boldsymbol{\nu}_i$ be the eigenvalues and eigenvectors of the covariance matrix, $COV(p_i^1)$, with the condition of $\lambda_1 \leq \lambda_2 \leq \lambda_3$. The change of curvature is a parameter of how much the surface formed by a point and its neighbourhood deviates from the tangential plane formed by $\boldsymbol{\nu}_2$ and $\boldsymbol{\nu}_3$. The ratio of the minimum eigenvalue and the sum of the eigenvalues approximates the change of geometric curvature,

$$M_{cc}(p_i^1) = \frac{\lambda_1}{\sum_{i=1}^{3} \lambda_i}. \tag{4}$$

Additionally, the geometric curvature, $M_{curv}(p_i^1)$, of a point can be estimated by the normal vectors of the point and its neighbourhood

$$M_{curv}(p_i^1) = \frac{1}{k}\sum_{j=1}^{k} ||\mathbf{n}_{p_i^1} - \mathbf{n}_{neighbour\{j,p_i^1\}}|| \tag{5}$$

where $\mathbf{n}_{p_i^1}$ and $\mathbf{n}_{neighbour\{j,p_i^1\}}$ are the normal vectors of p_i^1 and its jth neighbourhood, (Linsen, 2001). The change of curvature, $M_{cc}(p_i^1)$, can be expressed

$$M_{cc}(p_i^1) = \frac{1}{k}\sum_{j=1}^{k} |M_{curv}(p_i^1) - M_{curv}(p^1_{neighbour\{j,p_i^1\}})|. \tag{6}$$

Both methods can provide a good approximation to the change of curvature. However, the quality of estimation depends on how well the neighbourhood points are distributed. Since our algorithm is for the registration of unorganised point clouds, there is no guarantee that every point

of a cloud has a set of evenly-distributed neighbourhood points. This problem can be overcome by using the angle criterion between the neighbour points as Linsen (2001) did for the triangulation of point clouds. Furthermore, we could use a method in which each point has different number of neighbourhood pairs. The angle criterion has been used in this paper but using different number of neighbourhood for each point was not implanted. Therefore, a larger number of neighbourhood points has been used.

2.3 Algorithm description

The amount of data to process in order to find correspondence is very large, which limits the robustness of registration algorithms. The higher curvature points may have more valuable information than the lower curvature points since they could be edges or corners. Therefore, in the early stages of iteration, we only take into account higher curvature points and then, as iteration proceeds, lower curvature points also are included to improve the registration. Our method for the registration of three-dimensional, partially overlapping and unorganised point clouds without good a priori alignment can be briefly described as follows:

1. Find the k neighbourhood points of every point in C^1 and C^2. Estimate the geometric primitives of the points.

2. Take initial sample points, $p^1_{\{1,\cdots,n_{iter=i}\}}$, whose change of curvature is greater than $T^{iter=i}_{sample}$ where $n_{iter=i}$ is the number of sample in ith iteration.

3. Find corresponding points of $p^1_{\{1,\cdots,n_{iter=i}\}}$. p^2_j is the corresponding point of p^1_i if

$$\theta(p^1_i, p^2_j) \leq T^{iter=i}_{normal}$$
$$\beta(p^1_i, p^2_j) \leq T^{iter=i}_{cc}$$

   where $T^{iter=i}_{normal}$ and $T^{iter=i}_{cc}$ are the thresholds for the angle between the normal vectors and the difference in the changes of geometric curvature between the corresponding points, respectively.

4. Calculate the approximate transformation, $T_{iter=i}$, and transform C^1. Rotate the normal vectors of all points of C^1 as well.

5. Update the threshold values in order to apply a more strict criterion for determination of possible corresponding points as follow.

$$T^{iter=i+1}_{normal} = T^{iter=i}_{normal} - \Delta T_{normal}$$
$$T^{iter=i+1}_{cc} = T^{iter=i}_{cc} - \Delta T_{cc}$$
$$T^{iter=i+1}_{sample} = T^{iter=i}_{sample} - \Delta T_{sample}$$

6. Calculate the registration error, $\epsilon^{iter=i}$, which is defined as the rms distance of points and their corresponding surfaces in our method. If $\epsilon^{iter=i}$ is greater than threshold, then go to step 2. Otherwise stop the registration. In addition, if $\epsilon^{iter=i}$ is smaller than $T_{\epsilon_{CM}}$, for example, the average distance of a point from its neighbourhood, then Chen and Medioni's method is used since it converges quickly than Horn's algorithm does if the point clouds are close (Rusinkiewicz and Levoy, 2001). Otherwise Horn's method is used.

If the initial alignment is close to the correct one, only a small number of points need to be searched. Otherwise a large number of points must be searched in order to find correct correspondence of sample points. The optimal number of points being searched could be evaluated from the statistical properties of the distribution of registration error metric (Zhang, 1994). However, the distance distribution of the corresponding points is usually not a unimodal Gaussian but bimodal or multimodal distributions. Furthermore, good initial alignment is not assumed in the proposed method, it is difficult to remove outliers in the early stages of iteration. Therefore, a large number of points need to be searched in order to determine the correspondence of two point clouds.

2.4 Scale of selected corresponding points

The scale of selected corresponding points is usually assumed to be unity and this assumption is reasonable in most cases (Horn, 1987). It can be also used as the error metric to represent the quality of registration (Crosilla and Beinat, 2002). The scale can be interpreted as a parameter for the quality in the determination of the corresponding points. For example, if we have incorrect correspondence information, then the scale is not unity. The scale factor in kth iteration, $s_{iter=k}$, can be expressed as

$$s_{iter=k} = \frac{\sum_{i=1}^{n_{iter=k}} \mathbf{p}^1_i \cdot T_{iter=k}(\boldsymbol{CP}(p^1_i, C^2))}{\sum_{i=1}^{n_{iter=k}} ||T_{iter=k}(\boldsymbol{CP}(p^1_i, C^2))||^2} \quad (7)$$

where $T_{iter=k}$ is the calculated transformation of the kth iteration, $\boldsymbol{CP}(p^1_i, C^2)$ is the position vector of the corresponding point of p^1_i, and $n_{iter=k}$ is the number of samples in the kth iteration. Although the scale of corresponding points is unity, it does not guarantee that we have one-to-one matched correspondence. If the scale is much greater or smaller than unity, the calculated translation could be incorrect.

2.5 Threshold values

The list of threshold values used in the proposed method is shown in Table 1. $T^{iter=0}_{cc}$ and ΔT are the most important and critical thresholds. The other threshold values are not critical to the success rate of the proposed method, although they affect the robustness of the registration. It is difficult to state explicitly which values are the optimal values since they depend on dataset. Currently we are working on finding the optimal and generalized expressions for these thresholds. Our suggestions of $T^{iter=0}_{cc}$ and ΔT from the experiences with the proposed method are

$$T^{iter=0}_{cc} = \frac{< M^1_{cc} > + < M^2_{cc} >}{2} \quad (8)$$

$$\Delta T_{cc} = \sqrt[2]{< M^1_{cc} >^2_{rms} + < M^2_{cc} >^2_{rms}} \quad (9)$$

threshold	description
k	number of neighbourhood
$T_{sample}^{iter=0}$	initial sampling threshold for the change of curvature
$T_{cc}^{iter=0}$	initial threshold for the difference in changes of curvature
$T_{normal}^{iter=0}$	initial threshold for the angle between normal vectors
ΔT_{sample}	increment for $T_{sample}^{iter=k}$
ΔT_{cc}	increment for $T_{cc}^{iter=k}$
ΔT_{normal}	increment for $T_{normal}^{iter=k}$
$T_{\epsilon_{CM}}$	threshold for starting Chen and Medioni's method
T_{ϵ}	threshold for stopping the registration

Table 1: The threshold values are used in the propose method.

where $< M_{cc}^{i} >$ and $< M_{cc}^{i} >_{rms}$ are the mean and rms of the change of curvature of C^{i}.

3 EXPERIMENTAL RESULTS

Three examples were tested with the proposed method: a simulated point cloud and two real point clouds captured with two different laser scanners. All datasets have partially overlapped. The proposed method was implemented in C++ and tested on a PC with Intel Pentium III 450MHz and 516MB RAM. Our program is not yet optimized so there is room for improvement in terms of processing speed. For neighbourhood search, we used a kd-tree search library developed by Arya et al. (1998) and LAPACK (1999) was used for covariance analysis.

3.1 Simulated data

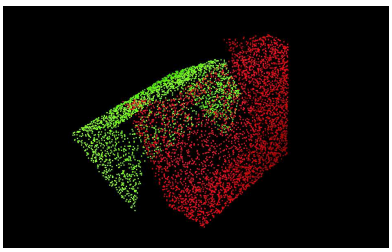


Figure 1: Before the registration of the point clouds of the parts of a cube

The simulated point clouds are parts of a cube, having dimensions of 1m×1m×1m, and partially overlapped. The number of points in the point clouds are 2640 and 4048. One point cloud was translated with (x,y,z)=(0.2m, 0.1m, 0.5m) and rotated 30° around z-axis from registered state as shown in Figure 1. Zero-mean Gaussian noise with various standard deviations was added independently to each point of the point clouds. In the case of zero standard deviation, i.e. no noise, all points in the overlapping region have exact corresponding points.

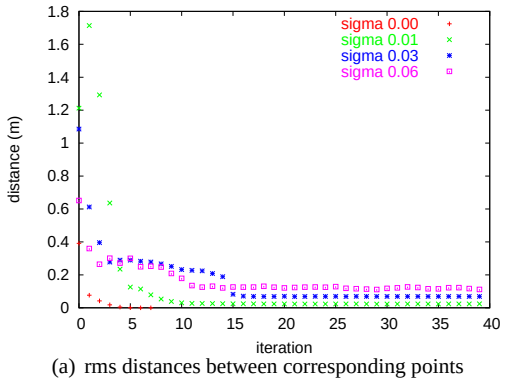
Many different error metrics have been defined to measure how well two point clouds are registered (Simon, 1996, Maas, 2000, Rusinkiewicz and Levoy, 2001). These include the change of rotation angles or translation, the distances between corresponding points, the distances between points and their corresponding surfaces, and so on. Whether the registration error, ϵ, is reasonable, too optimistic or pessimistic, depends mainly on the number of outliers that are used to register the point clouds. In addition, the redundancy of correspondence, the spatial density of data and the percentage of the overlapping regions are important factors. Two parameters that represent the error of registration were measured: the distances between corresponding points and the distances of points from their corresponding surfaces. Figure 2 shows these measures for the simulated dataset. As expected, more iterations are needed in order to minimise registration error, as more noise is added to the point clouds. The magnitude of the distances between corresponding points is about four times greater than the distances between points and their corresponding surfaces. It means that the success rate to find the correct point-to-point correspondence is much smaller than that to find the correct point-to-surface correspondence. This is not surprise if we consider that the test point clouds are parts of a cube, i.e. most of overlapping regions of the point clouds possess low curvature area. Therefore, we use the distance between point and its corresponding surface as the error metric of our method. Although we use this as error metric, the distance between corresponding points will still provide good information to increase the efficiency of our algorithm since we may remove outliers based on that information.

The scales of selected corresponding points in each iteration of the registration of simulated point clouds with various standard deviations of zero-mean Gaussian noise are shown in Figure 3. In early stage of registration, scales are much greater than unity since we do not have good a priori alignment. After about five iterations, all scales of the different levels of noise become approximately unity, which is a good indication of success in finding correspondences. However, there are some differences between the scales in the presence of noise as shown in Figure 3(b).

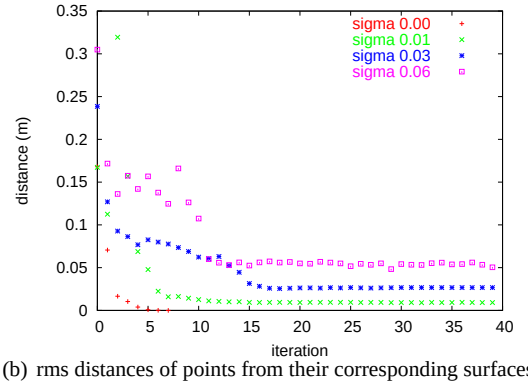
3.2 Real point clouds

The second example is the registration of two real point clouds from a Buddha statue (Ayuthaya, Thailand), scanned with Riegl LMS-Z210 that has angular sampling interval is 0.018° (Riegl, 2004). Figure 4 shows the point clouds as before and after registration using our method. The third example is a scene containing a building and trees measured by Mensi GS200 whose angular sampling interval is 0.0025° (Mensi, 2004). In this example, three point clouds are registered as shown in Figure 5.

The results of registration are listed in Table 2. In case of the simulated data without noise, the registration error after seven iterations is 0.04mm. In the cases of simulated point clouds with zero-mean Gaussian noise, registration errors are similar with the standard deviations of Gaussian noise. The execution time of $\sigma = 0.06$ is faster than the other cases. All registration errors of both simulated and

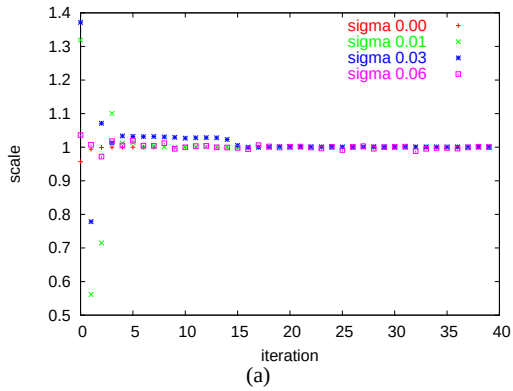


(a) rms distances between corresponding points

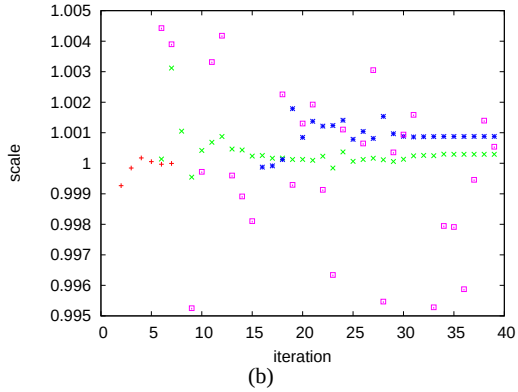


(b) rms distances of points from their corresponding surfaces

Figure 2: Two kinds of registration errors of simulated point clouds with different levels of noise. σ is the standard deviation of zero-mean Gaussian noise.

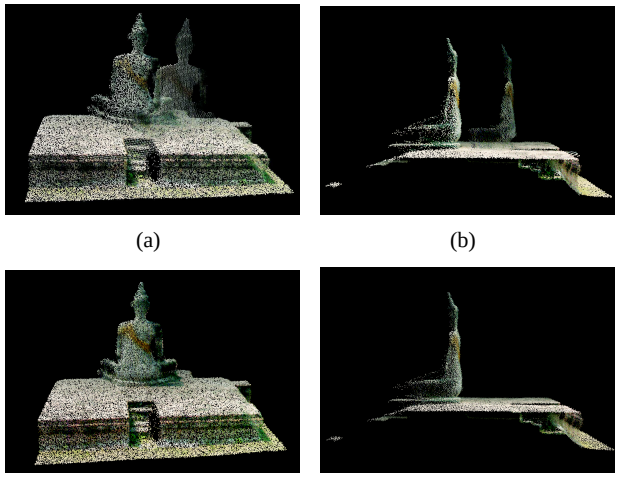


(a)



(b)

Figure 3: The scale of selected corresponding points in each iteration of the registration of simulated point clouds with zero-mean Gaussian error. (b) is the magnified figure of (a).



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

Figure 4: A Buddha statue scanned by Riegl LMS-Z210. (a) and (b) are before the registration. (c) and (d) are after the registration.

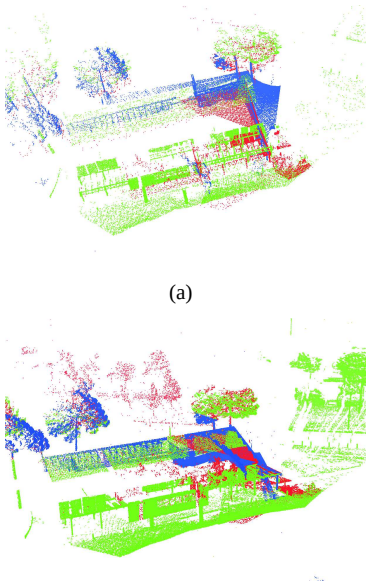
real point clouds are much smaller than the point spacings of point clouds defined as the average distance from a point from its neighbourhood. The registration errors of the two real point clouds are the order of centimetre. In the cases of the building and trees captured by the Mensi GS200, registration is successful as indicated by the registration error, ϵ, despite the difference of the point spacings of two point clouds being about the order of 10cm and the presence of many trees, which hinders the registration of the point clouds.

	n_1 n_2	k	i	t (sec)	ϵ (m)	d_1 d_2
Cube	2640	40	7	3.0	0.000040	0.119
σ = 0.0	4048					0.118
Cube	2640	40	39	21.0	0.00915	0.119
σ = 0.01	4048					0.118
Cube	2640	40	39	21.0	0.0267	0.119
σ = 0.03	4048					0.118
Cube	2640	40	39	16.0	0.0504	0.119
σ = 0.06	4048					0.118
Ayuthaya	39268	30	49	62.0	0.0235	0.043
	4393					0.061
building	139665	10	49	323.0	0.0388	0.194
(2+1)	217377					0.361
building	139665	10	49	602.0	0.0238	0.194
(2+3)	325870					0.371

Table 2: Results of experiments with simulated and real point clouds. n_i is the total number of points of point cloud C^i. k and i are the numbers of the neighbourhood of a point and total iterations, respectively. t and ϵ are the execution time and the registration error. d_i is the point spacing which is defined as the average distance of a point from its neighbourhood.

4 CONCLUSION

A method for the registration of two partially overlapping point clouds from different locations without good a priori alignment was proposed and tested with a simulated point cloud with different levels of Gaussian noise and two



(a)

(b)

Figure 5: A building and trees scanned by Mensi GS200. (a) and (b) are before and after the registration, respectively.

real point clouds from two different scanners. The distance from a point and the corresponding surface was used as the error metric of registration. The registration errors for real point cloud registration were the order of centimetre and that of a simulated dataset were similar with the standard deviations of zero-mean Gaussian noise.

Several ways are possible to improve our method. In terms of execution times, we can modify our method to use different neighbour points for each point depending on the distribution or the area of the region covered by the point and the neighbourhood. Regarding threshold values, the properties of the threshold values used in the proposed method can be investigated in order to provide criteria for the selection of the optimal threshold values. Furthermore, corner points of point clouds can be detected using geometric primitives that have used in the proposed method and they can be used as initial samples for the registration. In addition, the scale of corresponding points may be a good indication of the quality of sampling for registration.

5 ACKNOWLEDGEMENT

This research was supported by an Australian Research Council (ARC) Discovery grant DP0342887. The authors thanks to Mensi for provision of GS200 dataset.

REFERENCES

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D., 1999. LAPACK Users' Guide. Third edn, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Arya, S., Mount, D., Netanyahu, N. S., Silverman, R. and Wu, A. Y., 1998. An optimal algorithm for approximate nearest neighbour searching. Journal of the ACM (Association for Computing Machinery) 45, pp. 891–923.

Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-D shapes. IEEE Transactions on Pattern Recognition and Machine Intelligence 14(2), pp. 239–256.

Chen, Y. and Medioni, G., 1992. Object modelling by registration of multiple range. Image and Vision Computing 10(3), pp. 145–155.

Crosilla, F. and Beinat, A., 2002. Use of generalised Procrustes analysis for the photogrammetric block adjustment by independent models. The ISPRS Journal of Photogrammetry and Remote Sensing 56, pp. 195–209.

Haralick, R. M., Joo, H., Lee, C. H., Zhang, X., Vaidya, V. G. and Kim, M. B., 1989. Pose estimation from corresponding point data. IEEE Transactions on Systems, Man, and Cybernetics 19(6), pp. 1426–1446.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., 1992. Surface reconstruction from unorganized points. Computer Graphics 26, pp. 71–78.

Horn, B. K. P., 1987. Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America 4(4), pp. 629–642.

Linsen, L., 2001. Local versus global triangulation. Proceedings of EUROGRAPHICS 2001.

Maas, H. G., 2000. Least-sqaures matching with airborne laser scanning data in a TIN structure. The ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 33(3A), pp. 548–555.

Mensi, 2004. http://www.mensi.com/, accessed on 20th April 2004.

Riegl, 2004. http://www.riegl.co.at/, accessed 20th April 2004.

Rusinkiewicz, S. and Levoy, M., 2001. Efficient variant of the ICP algorithm. Proceedings of 3-D Digital Imaging and Modelling (3DIM).

Sharp, G. C., Lee, S. W. and Wehe, D. K., 2002. ICP registration using invariant features. IEEE Transactions on Pattern Recognition and Machine Intelligence 24(1), pp. 90–102.

Simon, D., 1996. Fast and Accurate Shape-Based Registration. PhD thesis, Robotics Institute, Carnegie Mellon University.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision 13(2), pp. 119–152.