Coarse point cloud registration by EGI matching of voxel clusters

Wang, Jinhu; Lindenbergh, Roderik; Shen, Yueqian; Menenti, Massimo

COARSE POINT CLOUD REGISTRATION BY EGI MATCHING OF VOXEL CLUSTERS

Jinhu Wang[a,b,][*] Roderik Lindenbergh[a], Yueqian Shen[a,c] and Massimo Menenti[a]

[a] Dept. of Geoscience and Remote Sensing, Delft University of Technology
Building 23, Stevinweg 1, Post Box 5048, 2628CN Delft, The Netherlands
(jinhu.wang, r.c.lindenbergh, y.shen-2, m.menenti)@tudelft.nl
[b] Key Laboratory of Quantitative Remote Sensing Information Technology
Academy of Opto-Electronics, Chinese Academy of Sciences
No. 9 Deng Zhuang South Road, HaiDian District, 100094 Beijing, China
[c] School of Earth Science and Engineering, Hohai University
No.1 Xikang Rd., 210098 Nanjing, China.

Commission V/5

KEY WORDS: Laser scanning, Point clouds, Voxels, Clustering, Eigenvalues, Registration

ABSTRACT:

Laser scanning samples the surface geometry of objects efficiently and records versatile information as point clouds. However, often more scans are required to fully cover a scene. Therefore, a registration step is required that transforms the different scans into a common coordinate system. The registration of point clouds is usually conducted in two steps, i.e. coarse registration followed by fine registration. In this study an automatic marker-free coarse registration method for pair-wise scans is presented. First the two input point clouds are re-sampled as voxels and dimensionality features of the voxels are determined by principal component analysis (PCA). Then voxel cells with the same dimensionality are clustered. Next, the Extended Gaussian Image (EGI) descriptor of those voxel clusters are constructed using significant eigenvectors of each voxel in the cluster. Correspondences between clusters in source and target data are obtained according to the similarity between their EGI descriptors. The random sampling consensus (RANSAC) algorithm is employed to remove outlying correspondences until a coarse alignment is obtained. If necessary, a fine registration is performed in a final step. This new method is illustrated on scan data sampling two indoor scenarios. The results of the tests are evaluated by computing the point to point distance between the two input point clouds. The presented two tests resulted in mean distances of $7.6mm$ and $9.5mm$ respectively, which are adequate for fine registration.

1 INTRODUCTION

The use of laser scanners for obtaining surface information of objects is increasing. Laser scanners provide dense 3D point clouds with high accuracy and versatile information about objects (Wehr and Lohr, 1999, Burtch, 2002, Large and Heritage, 2009, Vosselman and Maas, 2010). In general, different partially overlapping scans are acquired to fully cover a scene. To transform different scans into a common coordinate system, registration is a compulsory step in point cloud processing (Sharp et al., 2002, Bae and Lichti, 2004, Wang, 2013).

There is plenty of literature on solving the registration problem for 3D point clouds. The solutions can roughly be categorized into three main methods. First, manual registration, which uses common points from different scans (Zhao et al., 2005). However, when there are many scans it is labor intensive to manually conduct registration. Second, semi-automatic alignment, which is mainly performed by placing signalized targets within the scanner field of view. Those targets are tie-points between different scans (Vosselman and Maas, 2010). In most cases the targets can be identified from the point cloud data, however, in some situations human interaction is required to identify the targets (Zhang, 2015). Finally, fully automatic registration, when point clouds are transformed to the same coordinate system in a fully automatic way (Makadia et al., 2006, Bae and Lichti, 2008, Nüchter et al., 2010). The ICP algorithm is often employed to acquire fine registration (Chen and Medioni, 1991, Besl and McKay, 1992, Zhang, 1994). However, a prior coarse registration is needed to

initiate the ICP algorithm in order to ensure it converges to its global minimum (Sharp et al., 2002, Du et al., 2010).

A plethora of papers have provided different solutions to the problem of coarse transformation determination (Gressin et al., 2013). Most methodologies identify either feature based or keypoint based correspondences. The features include geometric primitives, higher level shape descriptors and special patterns. For example in (Beis and Lowe, 1999, Huber and Hebert, 2003), segments and curves are employed, while in (Dold and Brenner, 2006, Frome et al., 2004), local planes, spheres and cylinders are used as correspondence candidates. Another strategy is to represent 3D point clouds as meshes, and then conduct the alignment on the resulting models (Szeliski and Lavallée, 1996, Allen et al., 2003). The method proposed in this paper applies an existing shape descriptor on shapes derived by clustering voxels based on their dimensionality.

After features are extracted, the next steps are to match correspondences, reject false correspondences, minimize the model function to estimate the transformation coefficients and apply the transformation to the point clouds. While finding correspondences, false matches may occur as outliers. Those outliers have to be removed in order to achieve a valid coarse registration. Given the mathematical model of the transformation, the coefficients of the transformation can be determined by minimizing the distance between the matches. The most commonly used algorithm for outlier removal is RANSAC (Zuliani, 2008, Raguram et al., 2008). Another method is to estimate the transformation parameters in a least-squares procedure (Gruen and Akca, 2005, Grant et al., 2012). Once a coarse registration is achieved, fine registration is performed to obtain final results.

---
[*]Corresponding author

This paper is organized as follows. In Section 2, the overall methodology of the proposed method is presented. Consecutively in Section 3 the testing results of the method are given. Conclusions and discussions are in Section 4.

2   METHODOLOGY

The method proposed in this study is shown in Figure 1. The processing workflow starts with two input point clouds, *Source* and *Target*. Then four consecutive procedures are conducted on those point clouds, which are voxelization, dimension analysis, clustering similar voxel cells and constructing EGI features for all clusters. Voxelization is a procedure that re-samples point
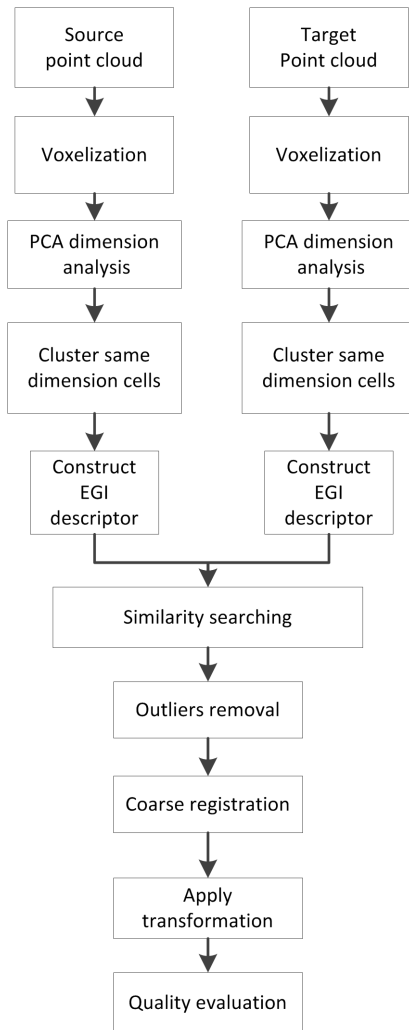

Figure 1: Overall methodology of the proposed coarse registration

clouds into voxel cells. Points inside each cell are analyzed and their dimension is obtained through PCA. Consecutively, cells that have the same dimension are clustered. Then the significant eigenvectors from those clustered cells are utilized to determine an EGI descriptor of each cluster. The similarity in the resulting EGI descriptor of the clusters from the two point clouds are compared to find correspondences. False correspondences are removed by RANSAC and coarse registration is achieved. The following step is fine registration employing the ICP algorithm. The final step is the quality evaluation of the registration results, which is conducted by computing the point-to-point distances between the two registered point clouds. The details of the method are given in the following sections.

2.1   Voxelization

The imported point clouds are re-sampled as voxel cells. As a *Pixel* is a 2D representation, a *Voxel* is a cube with a predefined edge length in 3D. The edge length also defines the resolution of the representation. Figure 2 is an illustration of a voxel cell in 3D. In this figure, P_{min} and P_{max} are the lower-left-front
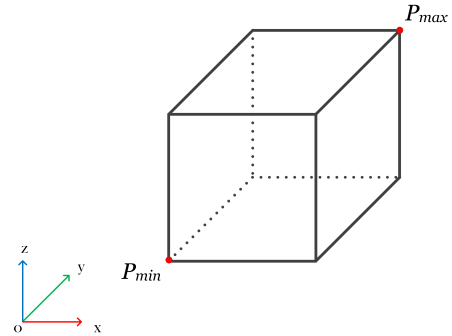

Figure 2: Voxel cell in 3D

and upper-right-back points respectively, which also defines the spatial extent of the voxel. The coordinate system used here is right-handed Cartesian.

The voxelization starts with obtaining the bounding boxes from the input point clouds. Then with a predefined voxel cell size, these bounding boxes are divided into voxels. Consecutively the points from the imported point cloud are assigned to the cells they fall in. To save memory costs, the data structure designed for a voxel cell consists of three entities: (i) a 3D index of the cell; (ii) a container storing the indices of the points that are within the cell; and (iii), coordinates of the 3D center of gravity of all the points inside this cell.

2.2   Voxel dimensionality analysis

The dimension of each voxel cell is determined by PCA as follows: Suppose $p_i = \begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^T$ are the coordinates of point p_i of point cloud P_s inside the voxel cell, then the center of gravity of all the points inside the cell is determined by Formula 1.

$$\overline{p} = \frac{1}{n} \sum_{i=1}^{n} p_i \tag{1}$$

Here $\overline{p}$ is the center of gravity and n is the number of points inside the voxel cell. Then the 3D structure tensor is defined by Formula 2.

$$R = \frac{1}{n} Q^T Q \tag{2}$$

Here R is the 3D tensor and $Q = (p_1 - \overline{p}, p_2 - \overline{p}, ..., p_n - \overline{p})^T$. R is a symmetric matrix that is decomposed as: $R = MIM^T$. Here M is a rotation matrix and I is a diagonal positive definite matrix. The elements of I are the eigenvalues of matrix R, and the column vectors of matrix M are the eigenvectors of matrix R. The eigenvalues are positive and are denoted by $\lambda_1, \lambda_2, \lambda_3$, and the corresponding eigenvectors are v_1, v_2, v_3. Suppose that $\lambda_1 > \lambda_2 > \lambda_3$. If $\lambda_1 >> \lambda_2$, then this voxel cell is defined as a linear, or $1D$ cell, and eigenvector v_1 is the main direction of the distribution of the points inside the cell. In this case, eigenvector v_1 is the significant eigenvector. if $\lambda_2 >> \lambda_3$, then the voxel cell is defined as a planar, or $2D$ cell. In this case eigenvector v_3, the normal vector of the plane, is the significant eigenvector; if

$\lambda_1 \approx \lambda_2 \approx \lambda_3$, then this cell is defined as a spherical, or 3D cell, and the cell has no dominant direction. Here $>>$ means *much larger*, and is implemented by a preset threshold. In this study, the linearity is examined first and then planarity. The thresholds are set to 10 and 20 respectively. After this procedure, all voxel cells have a geometric flag denoting its dimensionality and three eigenvalues and eigenvectors respectively as its properties.

2.3 Clustering voxel cells of the same dimension

In this step, face-connected voxel cells having the same dimension are clustered. In this work a modified 3D seed filling algorithm is employed to fulfill the processing. The workflow is shown in Figure 3. First voxel cells are traversed to if each voxel
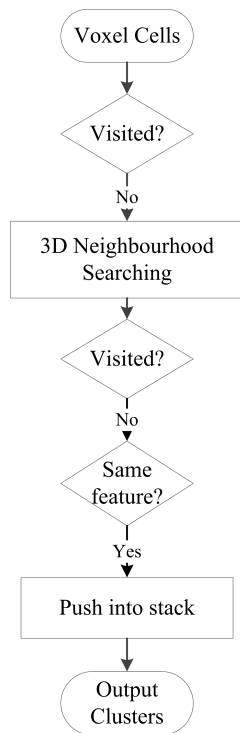

Figure 3: Modified 3D seed filling clustering of voxel cells

cells is visited, visited, if not, then its 26 neighbor cells are searched. Then for all the non-visited neighbors, if it has the same dimension as the current cell, then the index of this cell is pushed into a cluster stack and this cell is labeled as visited. This same processing is performed for all the cells until all voxel cells are visited.

2.4 The EGI descriptor of a voxel cluster

In this work the Extended Gaussian Image (Horn, 1984) feature is modified to identify correspondences among clustered voxel cells from both imported scans. The significant eigenvectors are used as identified in the dimensionality analysis in Section 2.2. Spherical cells do not have a dominant direction, therefore only linear and planar cells are considered in this work. Approximate sphere used to assign significant eigenvectors to is defined the *Eigen-Sphere* of a cluster. A full sphere is approximated by an icosahedron, as illustrated in Figure 4.

This is a uniform icosahedron, as the distances of its vertices to the origin O are all equal to 1. An icosahedron can be further tessellated by subdivision of each triangular surface into four triangles by connecting the center points of each edge.
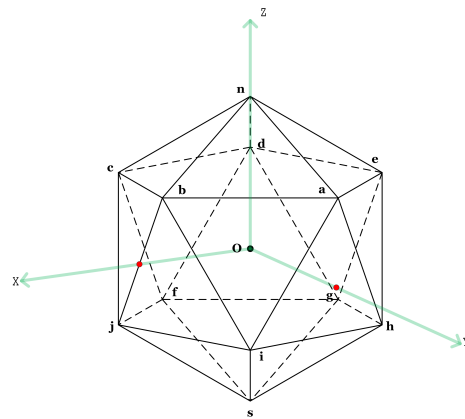

Figure 4: Standard icosahedron used to construct a modified EGI descriptor of a cluster

The next step of the procedure is to assign the significant eigenvectors of each voxel in the cluster to the triangles on the boundary of the icosahedron. In Figure 5, v_1, v_2, v_3 are the three ver-
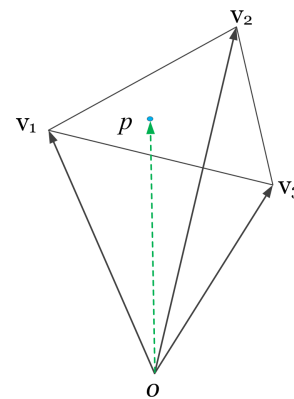

Figure 5: Assigning eigenvectors onto a triangular surface of an icosahedron

tices of a triangle on the boundary of the icosahedron, and vector p is intersecting this triangle. If vector $\vec{p}$ is intersecting with the triangle, then the coefficients k_1, k_2, k_3 in the linear combination in Equation 3 must be all positive (Preparata and Shamos, 1985).

$$\vec{x} = k_1 v_1 + k_2 v_2 + k_3 v_3 \qquad (3)$$

Therefore Equation 3 can be decomposed to determine the coefficients by Equation 4.

$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^{-1} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad (4)$$

Here $v_1 = (x_1 \quad x_2 \quad x_3)_T$ are the coordinates of vertex v_1 and x, y, z are the coordinates of vector $\vec{p}$. The *Eigen-Sphere* feature is constructed for each cluster by assigning the significant eigenvectors of all voxels in the cluster to its *Eigen-Sphere* by determining for each eigenvector which triangle of the icosahedron it intersects using Equation 4. That is, for a linear cluster, only the voxel eigenvectors corresponding to the biggest eigenvalues are

assigned and for a planar cluster the normal vectors of its voxels are assigned.

Note that the vectors obtained by PCA are not direction definite, for example, vector v_0 and its antipodal vector $-v_0$ are both applicable. In this work, for the purpose of symmetrical concern, both the vector and its antipodal vector are assigned to the *Eigen-Sphere*.

An icosahedron has a total of 60 symmetry transformations (Conway et al., 2008). In this work, both a significant PCA vector and its opposite vector are assigned to the icosahedron surfaces, therefore only 30 symmetries have to be considered in practice. Similarity is only determined between clusters containing voxels of the same dimension, i.e. linear clusters with linear clusters and planar clusters with planar clusters respectively. The similarity between a pair of *Eigen-Sphere* descriptors, named as *source* and *target*, is defined by Equation 5.

$$S_k^{s,t} = \sum_{i=1}^{20}(N_i^s - Ni^t)^2 \qquad (5)$$

Here $S_k^{s,t}$ is the similarity between *source* and *target* for the *k-th* symmetry. N_i^s and Ni^t are the number of assigned significant eigenvectors intersecting on the *i-th* triangular face of the icosahedron. Note that there are 20 faces on an icosahedron, $i = 1, 2, ..., 20$. There are 30 symmetry transformations in total, so $k = 1, 2, ..., 30$.

The symmetry transformation that minimizes similarity among the total of 30 cases is defined as the cluster pair similarity between two clusters from two scans both containing voxels of the same dimension (linearity or planarity). For all linear clusters in the source scan, the similarity of the cluster is determined to all linear clusters in the target scan. The pair with optimal cluster-pair similarity is considered a cluster match. Similarly, cluster matches between planar clusters are determined.

2.5 Remove outliers and Coarse registration

To deal with false correspondences in the list of cluster matches, RANSAC is employed. In (Besl and McKay, 1992), a Least Squares solution of 3D rigid registration of point sets is provided. The goal of the rigid registration between two 3D point sets is to find an optimal transformation consisting of a rotation and a translation. Equation 6 defines the model of 3D rigid registration.

$$(R, t) = \underset{R,t}{\arg\min} E(\mathbf{R}, \vec{t})$$
$$= \underset{R,t}{\arg\min} \sum_{i=1}^{N}[(\mathbf{R} \cdot C_i^s + \vec{t}) - C_i^t]^2 \qquad (6)$$

Here $E(\mathbf{R}, \vec{t})$ is an error function, $\mathbf{R}$ is the rotation matrix and $\vec{t}$ is the translation vector. N is the number of correspondences, C_i^s and C_i^t are the obtained *i-th* pair of correspondences, (R, t) are the determined transformation parameters.

False correspondences result in a larger error function in Equation 6. Before possible outliers can be removed with RANSAC, first the number of iterations and the threshold of tolerance have to be set according to the data properties. Moreover, to generate a more accurate coarse registration, the transformation parameters from the best k pairs of correspondences are averaged as the final parameters.

2.6 Fine registration and results evaluation

Taking the coarse registration from Section 2.5 as initial result, the ICP algorithm is applied to perform fine registration of the point clouds. The ICP algorithm has three steps: (i) from *Source* P^s point cloud, searching the closest point in *Target* point cloud P^t, and then compute the transformation parameters $(\mathbf{R}_m, \vec{t}_m)$; (ii) apply the transformation and obtain a new iteration of point $P_{m+1}^t = \mathbf{R}_m \cdot P_m^t + \vec{t}_m$; and (iii) terminate the iterations when the change in mean-square error is smaller than a preset threshold or the iterations exceed a predefined number of runs. Meanwhile the new transformation is applied to the point cloud. The evaluation of the registration results is performed by computing the point-to-point distance between the *Source* and *Target* point clouds.

3 RESULTS AND VALIDATION

The proposed method is evaluated on two data sets. The first pair of scans is created by applying a transformation on one scan. The second data set consists of two scans obtained from different positions. Both point cloud data sets used in this work were scanned by a Leica C10 TLS scanner.

3.1 Data description

The first data set samples an indoor scenario and the original point cloud is shown in Figure 6.
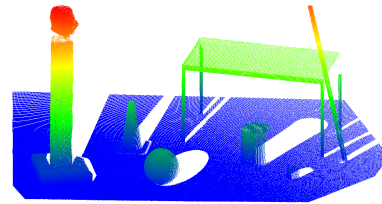

Figure 6: The original point cloud obtained with a Leica C10 TLS scanner.

The data set has 314,929 points and the average point density is around 50,000 points per square meters. In the scene, there is a table, a ball, a traffic cone and a cylindrical bin. The original point cloud is first rotated by 23 degrees about its z axis and translated by $50cm$ in all three Cartesian directions to generate a new point cloud for pair-wise registration. The original and the newly generated point cloud are regarded as *Source* and *Target* point clouds respectively in the following sections.

3.2 Voxelization results

The voxelization of the point cloud is illustrated in Figure 7. In this voxelization, the voxel cell size used is $15cm$. Smaller voxel size will result in finer cluster results but will also increase computation time.

3.3 PCA geometry analysis

The dimensionality of the cells is illustrated in Figure 8. In the figure, the point cloud is in gray, while the red and green cells represent linear and planar cells respectively. The cells in blue are spherical cells. In this work, linearity of a cell is first determined. A cell is regarded as a linear cell if the largest eigenvalue is at least 10 times larger than the second eigenvalue, that is when $\frac{\lambda_1}{\lambda_2} > 10$. If a cell is not a linear cell, then the planarity is tested.
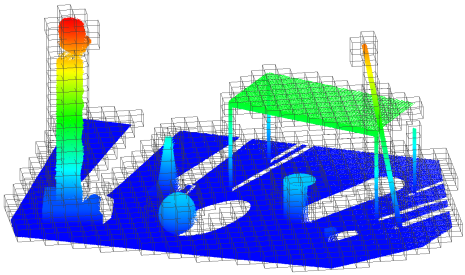
Figure 7: Voxelization of the point cloud with 15 cm voxels

A cell is considered as a planar cell if the smallest eigenvalue is 20 times smaller then the second largest eigenvalue, i.e. when $\frac{\lambda_2}{\lambda_3} > 20$. If a cell does not meet the above two criteria, then it will be considered as a spherical cell.
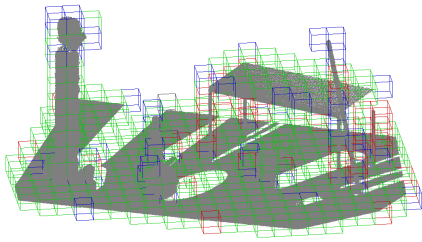

Figure 8: Geometric features of the voxel cells

3.4 Clustering voxel cells

After obtaining the properties in Section 3.3, connected cells having the same geometric properties are clustered to form voxel cell clusters. Figure 9 shows the clustering results.
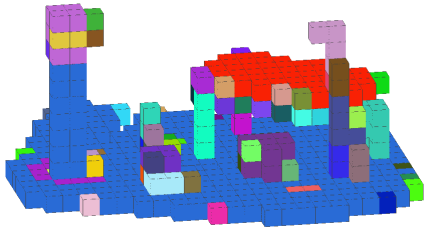

Figure 9: Clustering connected cells of the same dimensionality

In the figure, random colors are designated to different clusters. However, there are also clusters consisting of only a few voxels. In the so-called majority step, clusters consisting of 1,2, or 3 voxels are merged to the biggest neighboring cluster that have the same dimensionality. The refined clustering results are illustrated in Figure 10. Compared with Figure 9, all one-cell clusters are merged to adjacent bigger clusters.

3.5 Construct Eigen-Sphere descriptor

The *Eigen-Sphere* descriptor of each cluster is constructed and stored in its data structure. For example, the *Eigen-Sphere* descriptor of the table surface and the sphere in Figure 6 are shown in Figure 11. The colors of the triangles denote the number of the vectors that intersect the triangle.

In the figure, the radial line segments denote the significant eigenvectors of each cell in a cluster and the surfaces are colorized by
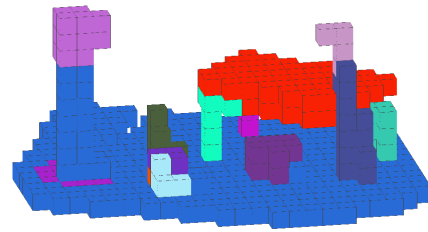

Figure 10: Refined clustering results of connected cells of the same geometric dimensionality
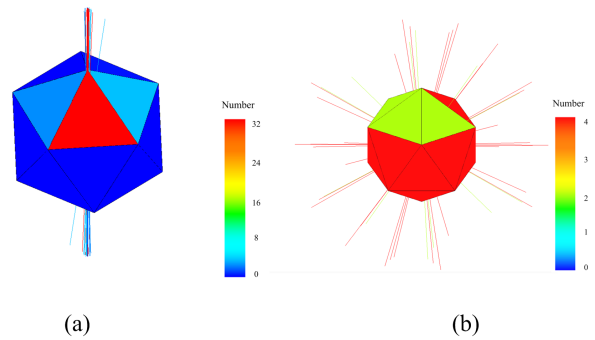

(a)   (b)

Figure 11: The *Eigen-Sphere* features of the table surface and the sphere

the number of significant eigenvectors that intersect each triangle. For the EGI descriptor of a planar surface, as depicted by Figure 11 (a), most of the significant eigenvectors point in the direction of the normal of that plane. For a spherical surface, in Figure 11 (b), the eigenvectors are distributed uniformly on the *Eigen-Sphere* surface, as different voxels correspond to different planar like patches at different sites of the sphere.

In this test, only planar clusters that have more than 20 cells were considered for correspondence matching. There were 7 pairs of matching clusters found as correspondences. Outliers in those matches were removed by the RANSAC algorithm with the iteration number and distance threshold set to 20 and $3cm$ respectively. Finally 4 true matches were found as illustrated in Figure 12.
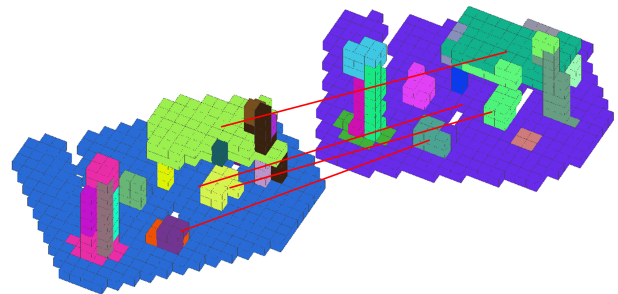

Figure 12: True matches from cluster-based EGI feature

3.6 Coarse registration

The result of the coarse registration is illustrated in Figure 13. The point to point distance of the registration is given in Figure 14, in which the points are colorized by the distance towards the other point cloud. The mean of the point to point distances

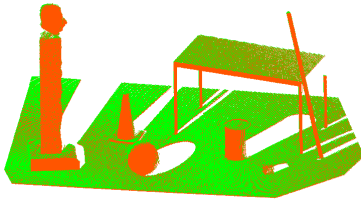is $7.6mm$, the std.dev. is $5.3mm$ and the median is $6.5mm$ respectively.

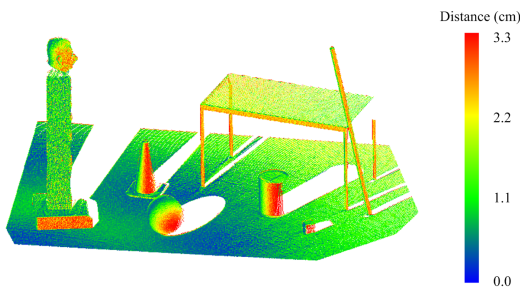Figure 13: Coarse registration results of the pair of point clouds

Figure 14: Distances between the scans after coarse registration

3.7 Fine registration

Taking the coarse registration in Section 3.6 as starting point of the ICP algorithm, the fine registration is conducted on the two point clouds. The fine registration results are given in Figure 15. The mean of the point to point distances of the source and target
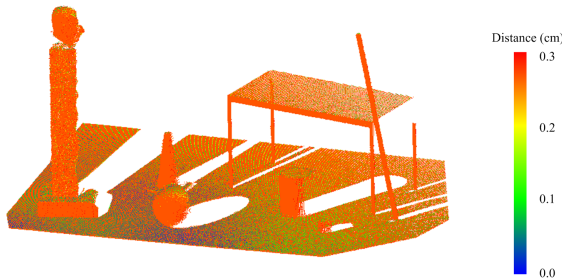

Figure 15: Distances after the fine registration

is 0.57 mm, the std.dev. is 0.18 mm and the median is 0.26 mm respectively. This results demonstrated that the coarse registration is good enough to ensure a fine registration.

3.8 Testing on another data set

To further verify the capability of the proposed coarse registration method, it is also tested on point clouds captured at different positions and of varying local point density. In this scenario, a wall was scanned by a Leica C10 TLS scanner from two different positions. The two point clouds contain 884,923 and 873,603 points respectively. In this test, the cell size was set to $10cm$ and similar processing was conducted on those point clouds. The dimension thresholds were set to 15 and 20 for linearity and planarity respectively. In total 8 true matches were found from 12 candidates using RANSAC with 20 iterations and $3.5cm$ threshold. The coarse registration result is illustrated in Figure 16. The mean and the std.dev. of the point to point distances of the two
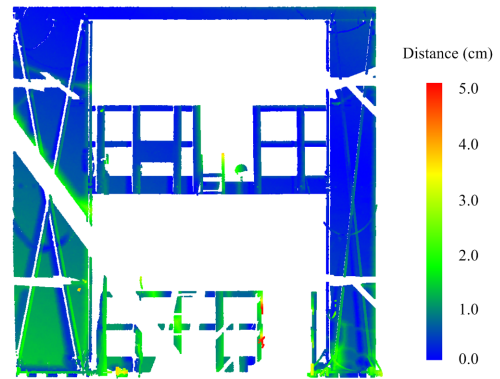

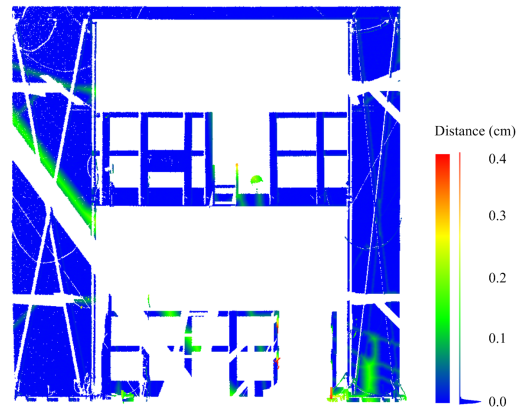Figure 16: Point to point distance of the coarse registration

Figure 17: Point to point distance of the fine registration

coarsely registered point clouds are $9.5mm$ and $6.7mm$ respectively.

The result of the fine registration of the point clouds is illustrated in Figure 17. The mean and std.dev. are $2.8mm$ and $1.7mm$ respectively. As can be distinguished in Figure 16 and Figure 17, the points that have bigger point to point distances correspond to actual differences between the two input point clouds. The two point clouds are obtained from two different stations and some parts of the facade are occluded. The histogram given right of the color bar in Figure 17 shows that the majority of the points in blue have small cloud to cloud distances.

In this test, the two point clouds were manually cropped and only the overlapping area was considered. This ensures the quality and efficiency of the method. The points in the non-overlapping area would introduce extra computation time and even errors. In addition, noise and outliers are removed by using an local neighborhood filtering algorithm (Rusu and Cousins, 2011). The results in this paper are presented as a proof of concept. To make this method work in an automated way on arbitrary input scans with only partial overlap is currently still considered challenging by the authors.

4 CONCLUSIONS

This paper presents a pair-wise coarse registration method for point clouds based on voxel cell clusters. First the input point clouds are voxelized and the dimension of each voxels is determined. Then voxel cells of the same dimensionality are clustered and EGI descriptors were constructed to obtain correspondences. The RANSAC algorithm was employed to remove out-

liers in the correspondences. The feasibility of the presented automatic coarse registration method is proven on two pairs of point clouds sampling indoor scenarios. The quality of the registration results are evaluated by the point to point distances of the registered point clouds. Some of the weaknesses are discussed in the results section. More data sets still need to be tested to verify the robustness of the presented method.

5 ACKNOWLEDGMENTS

REFERENCES

Allen, B., Curless, B. and Popović, Z., 2003. The space of human body shapes: reconstruction and parameterization from range scans. ACM Transactions on Graphics 22(3), pp. 587–594.

Bae, K. H. and Lichti, D. D., 2004. Automated registration of unorganised point clouds from terrestrial laser scanners. Update 35, pp. 222–227.

Bae, K. H. and Lichti, D. D., 2008. A method for automated registration of unorganised point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 63(1), pp. 36–54.

Beis, J. and Lowe, D., 1999. Indexing without invariants in 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(10), pp. 1–30.

Besl, P. and McKay, N., 1992. A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), pp. 239–256.

Burtch, R., 2002. Lidar principles and applications. Current pp. 1–13.

Chen, Y. and Medioni, G., 1991. Object modelling by registration of multiple range images. Proceedings of the 1991 IEEE International Conference on Robotics and Automation CH2969(4), pp. 2724–2729.

Conway, J. H., Burgiel, H. and Goodman-Strauss, C., 2008. The symmetries of things. Wellesley, Mass.

Dold, C. and Brenner, C., 2006. Registration of terrestrial laser scanning data using planar patches and image data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36, pp. 78–83.

Du, S., Zheng, N., Ying, S. and Liu, J., 2010. Affine iterative closest point algorithm for point set registration. Pattern Recognition Letters 31, pp. 791–799.

Frome, A., Huber, D., Kolluri, R., Bülow, T. and Malik, J., 2004. Recognizing Objects in Range Data Using Regional Point Descriptors. Computer Vision - ECCV 2004 3023, pp. 224–237.

Grant, D., Bethel, J. and Crawford, M., 2012. Point-to-plane registration of terrestrial laser scans. ISPRS Journal of Photogrammetry and Remote Sensing 72, pp. 16–26.

Gressin, A., Mallet, C., Demanté, J. and David, N., 2013. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. ISPRS Journal of Photogrammetry and Remote Sensing 79, pp. 240–251.

Gruen, A. and Akca, D., 2005. Least squares 3D surface and curve matching. ISPRS Journal of Photogrammetry and Remote Sensing 59(3), pp. 151–174.

Horn, B. K. P., 1984. Extended Gaussian Images. Proceedings of the IEEE 72(12), pp. 1671–1686.

Huber, D. F. and Hebert, M., 2003. Fully automatic registration of multiple 3D data sets. Image and Vision Computing 21(7 SPEC.), pp. 637–650.

Large, A. R. and Heritage, G. L., 2009. Laser Scanning Evolution of the Discipline. Laser Scanning for the Environmental Sciences pp. 1–20.

Makadia, A., Patterson, A. and Daniilidis, K., 2006. Fully Automatic Registration of 3D Point Clouds. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Volume 1 CVPR06 1(c), pp. 1297–1304.

Nüchter, A., Elseberg, J., Schneider, P. and Paulus, D., 2010. Study of parameterizations for the rigid body transformations of the scan registration problem. Computer Vision and Image Understanding 114(8), pp. 963–980.

Preparata, F. P. and Shamos, M. I., 1985. Computational geometry: an introduction. Springer.

Raguram, R., Frahm, J. M. and Pollefeys, M., 2008. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5303 LNCS(PART 2), pp. 500–513.

Rusu, R. B. and Cousins, S., 2011. 3D is here: Point Cloud Library (PCL). In: Proceedings - IEEE International Conference on Robotics and Automation.

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

Szeliski, R. and Lavallée, S., 1996. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. International Journal of Computer Vision 18(2), pp. 171–186.

Vosselman, G. and Maas, H.-G., 2010. Airborne and Terrestrial Laser Scanning. Whittles Publishing.

Wang, J., 2013. Block-to-point fine registration in terrestrial laser scanning. Remote Sensing 5(12), pp. 6921–6937.

Wehr, A. and Lohr, U., 1999. Airborne laser scanningan introduction and overview. ISPRS Journal of Photogrammetry and Remote Sensing 54(2-3), pp. 68–82.

Zhang, M., 2015. Accurate sphere marker-based registration system of 3d point cloud data in applications of shipbuilding blocks. Journal of Industrial and Intelligent Information 3(4), pp. 318–323.

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

Zhao, W., Nister, D. and Hsu, S., 2005. Alignment of continuous video onto 3D point clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), pp. 1305–1318.

Zuliani, M., 2008. RANSAC for Dummies. Citeseer 25(2), pp. 164–176.