Point Cloud Registration Using Intensity Features

Chien-Chou Lin,[1,2*] Wei-Lung Mao,[3**] and Ting-Lun Hu[1]

[1]Dept. of Computer Science and Information Engineering, National Yunlin University of Science & Technology,
No. 123, University Rd., Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C.
[2]Intelligence Recognition Industry Service Research Center, National Yunlin University of Science and Technology,
No. 123, University Rd., Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C.
[3]Dept. of Electrical Engineering and Graduate School of Engineering Science and Technology,
National Yunlin University of Science & Technology,
No. 123, University Rd., Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C.

In this paper, a registration method for extending point clouds is proposed. The proposed method merges several point clouds to increase the vertical field of view (FOV). However, the most popular alignment algorithm, iterative closest point (ICP), fails to extend point clouds that are captured with varying heights when most points are similar. The main issue is the tyranny of the majority, in which ground points and wall points dominate the registration result of ICP. Instead of using all points of point clouds, the proposed method only uses the intensity features to find the transformation matrix between two point clouds and then transforms the target point cloud to the coordinate system of the source point cloud. Upon merging the two point clouds, the vertical FOV can be extended. In a simulation, the proposed algorithm scans the source and the target with fixed position and varying height using a light detection and ranging (LiDAR) (Velodyne VLP-16 mounted on a tripod). The simulation result shows that the average error of alignment of the proposed system is less than 16 cm in a 6×6 m^2 meeting room, and the average error of alignment of the proposed system using a premeasured height for compensation is less than 12 cm.

1. Introduction

Light detection and ranging (LiDAR) is an important method for 3D object reconstruction. With the drop in price, LiDARs are becoming widely used sensors and the point cloud obtained by LiDAR is also widely used in many state-of-the-art approaches. Recently, the reconstruction of 3D objects/scenes has been an important research topic in computer vision. In general, applications of 3D reconstruction can be divided into two categories: one is reconstructing scene environments, e.g., building interiors, street views, or natural scenes. The other is constructing a single object, e.g., an item of furniture, a vehicle, or a factory product. For digital archive application, the details of the interior and exterior of historical buildings have to be scanned and

modeled for repairing, maintenance, and restoration. Usually, 3D scene reconstruction involves combining a stream of point clouds to establish a city scene. The stream of point clouds is captured by a LiDAR device mounted on a mobile platform or installed on a flight device to build city models and rugged ground surfaces.

A LiDAR device is an optical remote sensing device that uses laser light to measure the distance of a target. LiDAR can measure distances with high accuracy, identify the shape of an object, and establish the surrounding 3D geographic information model. It has the advantages of high-volume range measurement, high accuracy, high discrimination, and robustness to varying illumination. Generally, the horizontal field of view (FOV) is 360 degrees, but the vertical FOV depends on the type of LiDAR device. A LiDAR device with a wide vertical FOV and high vertical angle resolution is very expensive. For example, the cost of a 64-layer LiDAR device is ten times that of a 16-layer LiDAR device. Usually, applications of a LiDAR device depend on its vertical FOV and vertical angle resolution since the distance between the two scanning layers is large at long distances. A sparse point cloud cannot completely present the appearance of an object, resulting in a decrease in the degree of recognition and an increase in the difficulty of subsequent data processing and analysis.

To increase the vertical FOV of a LiDAR device for scanning the interior of a historical building, several point clouds captured at a fixed position and varying heights are aligned and merged together to form a point cloud with a wider vertical FOV.

In this paper, the proposed registration method only uses intensity features to improve the accuracy of extending point clouds by the iterative closest point (ICP). The proposed algorithm scans the source and the target with a fixed position and varying height, extracts the intensity features from both point clouds and then derives a better approximate transformation between the two point clouds by ICP. This paper consists of five sections. First, the motivation of this research is introduced. In Sect. 2, the related works are reviewed. The detail of the proposed method is given in Sect. 3. A simulation and conclusion are given in Sects. 4 and 5, respectively.

2. Related Works

Many algorithms[1–5] for 3D reconstruction have been proposed. The key to 3D reconstruction is aligning several point clouds with partial overlapping into a common coordinate system. The major task of alignment is to find the optimal transformation between the point clouds. By merging several aligned point clouds, the 3D environment or the 3D object can be reconstructed. Depending on the searching schemes used in alignment, the existing methods can be categorized into global-registration-based methods and local-registration-based methods. The global-registration-based methods use all points in the source to find the corresponding points in the target and then compute the approximate transformation matrix. In Ref. 1, Cordón *et al.* used the evolution algorithm to find the transformation and the gene algorithm used by Silva *et al.*[2] The disadvantage of this type of method is that its computational cost is too high. Another kind of local registration method is more commonly used in most studies and it usually uses the features of a 3D model, e.g., curves or planes, to find the corresponding features.[3–5]

However, the most popular method for alignment, the ICP algorithm, has several varieties. Some ICP-based approaches[6,7] use corresponding point pairs to find the transformation. Others[8–17] use corresponding features, e.g., planes or curves, to find the transformation of 3D scenes or other features between the reference and the source. Thus, ICP can be categorized into both the above types.

ICP is reviewed as follows. The ICP algorithm,[6] proposed in 1992 by Besl and McKay, consists of several steps: finding the closest point in the data shape for each point in the model shape, then estimating the rotation and translation by a mean squared error (MSE) cost function and transforming the data shape with respect to the obtained rotation and translation. Finally, the best result of alignment is obtained by iterating the above process until the transformation is negligible.

Let $A(x_j, y_j, z_j)$ be a point in A with m points and $B(x_k, y_k, z_k)$ be a point in B with n points. The MSE is defined as the summation of the Euclidean distances of the closest point with respect to all points in A.

When deriving the transformation, the cross-covariance matrix (CCM), the mean product of the closest point vectors, is used. **CCM** is defined as follows:

$$\boldsymbol{CCM} = \frac{1}{m}\sum_{i=1}^{m}\left(\begin{bmatrix} x_i^A - x_C^A \\ y_i^A - y_C^A \\ z_i^A - z_C^A \end{bmatrix} \cdot \begin{bmatrix} x_i^C - x_C^C \\ y_i^C - y_C^C \\ z_i^C - z_C^C \end{bmatrix}^{\mathrm{T}} \right), \tag{1}$$

where $\begin{bmatrix} x_C^A & y_C^A & z_C^A \end{bmatrix}^{\mathrm{T}}$ is the center point of *A*. **P**, a 3×3 matrix representing the difference between **CCM** and the its transpose, can be written as

$$\boldsymbol{P} = \boldsymbol{CCM} - \boldsymbol{CCM}^{\mathrm{T}}. \tag{2}$$

A quaternion matrix **Q** is defined as

$$\boldsymbol{Q} = \begin{bmatrix} trace(CCM) & r \\ r^{\mathrm{T}} & (CCM + CCM^{\mathrm{T}}) - trace(CCM) \times I \end{bmatrix}, \tag{3}$$

where **I** is the identity matrix and $r = \begin{bmatrix} P_{23} & P_{31} & P_{12} \end{bmatrix}$ is a row vector. Let $\vec{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}$ be an eigenvector of **Q**. Using the quaternion representation of rotation, the rotation matrix can be written as

$$\boldsymbol{R}_q = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 - q_1 q_4) & 2(q_2 q_4 - q_1 q_3) \\ 2(q_2 q_3 + q_1 q_4) & q_1^2 + q_3^2 - q_2^2 - q_4^2 & 2(q_3 q_4 - q_1 q_2) \\ 2(q_2 q_4 - q_1 q_3) & 2(q_3 q_4 + q_1 q_2) & q_1^2 + q_4^2 - q_2^2 - q_3^2 \end{bmatrix}. \tag{4}$$

Therefore, the translation vector is

$$
\boldsymbol{T} = \begin{bmatrix} x_C^C \\ y_C^C \\ z_C^C \end{bmatrix} - \boldsymbol{R}_q \begin{bmatrix} x_C^A \\ y_C^A \\ z_C^A \end{bmatrix}.
\tag{5}
$$

After an iteration of ICP, MSE is recalculated to determine whether the next iteration is necessary. If the change of MSE is negligible, the optimal transformation is obtained and then ICP stops.

3.   Alignment Using Intensity Features

ICP, as reviewed in the previous section, tends to align large parts of two point clouds with high similarity. However, the points of the ground and the wall dominate the registration result owing to the tyranny of the majority. Thus, ICP cannot be used for merging another point cloud with a similar extra part to extend the original point cloud. Instead of using all points of point clouds, the proposed method only uses the intensity features for ICP.

The proposed algorithm, as shown in Fig. 1, consists of five steps: (1) scanning the source point cloud with the initial position and height, (2) scanning the target point cloud with fixed position and different heights, (3) extracting the intensity features from both point clouds, (4) deriving the transformation matrix between the two point clouds by ICP, and (5) transforming the target point cloud into the coordinate system of the source point cloud and extending the source point cloud. The detail of the proposed algorithm is introduced in this section.
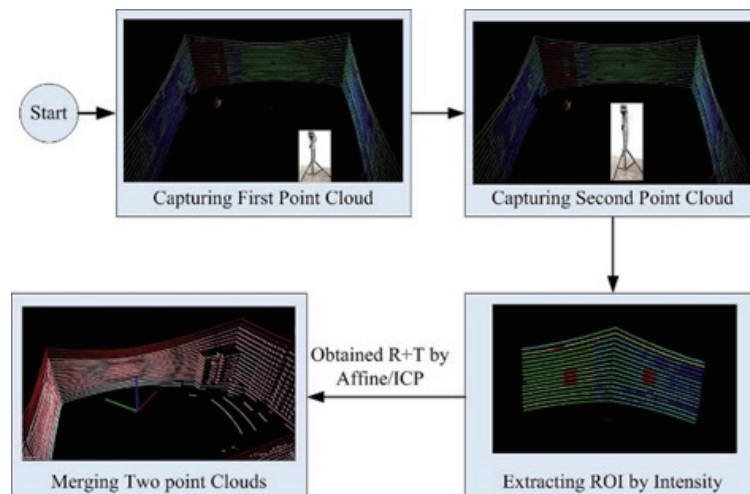

Fig. 1.   (Color) The proposed algorithm consists of five steps: (1) scanning the source point cloud, (2) scanning the target point cloud with different heights, (3) extracting the intensity features, (4) deriving the transformation matrix by ICP, and (5) transforming the target point cloud and extending the source point cloud.

3.1　Scanning the source point cloud and the target point cloud

The goal of this paper is to extend the vertical FOV of a point cloud by merging several point clouds. The source point cloud and target point cloud are individually captured in the first and second steps. In the second step, the position of the LiDAR device is fixed but its height is increased to scan the extra area. Since the laser of the LiDAR device is invisible, the exact scanned position is unknown. Therefore, two black patterns are put up on two non-coplanar walls for separate calibration. In contrast to the white walls, for which the intensity of reflected light is strong, the black patterns made of a nonwoven fabric have low reflection and are easy to extract in point clouds with intensity information. The first scanned point cloud is shown in Fig. 2, in which blue represents stronger intensity and red represents weaker intensity.

Note that if the two patterns are put on the same wall, the obtained transformation is singular. In other words, the translation vector and rotation matrix between the two point clouds are uncertain. Thus, the feature blocks are put up on two non-coplanar walls.

3.2　Extracting the intensity features

In general, when a LIDAR device captures a scanned scene, the raw data includes the distances and signal intensity values of scanned points. The reflection intensity will depend on the material of the object. When the laser light hits a smooth and nontransparent plane, the intensity of the reflection is greater than that on a plane with a matte surface. Therefore, if some blocks with low-reflectivity materials are set in one scene, the characteristics of the scanned data are used as features for better calibration.

In our experiments, the non-woven fabric is used as the feature block when scanning and the range of reflection intensity values is measured in advance. The feature size used for scene scanning is 30×30 cm^2. Because the intensity value of the feature points is much lower than that of the surrounding points, the calibration pattern can be easily extracted by the predetermined thresholds. The classification of points can be expressed as

$$P_i \in S_{CP}, \text{ if } I_{min} \leq I_{P_i} \leq I_{max}, \tag{6}$$

where S_{CP} is the point set of the calibration pattern and I_{max} and I_{min} are the predetermined intensity thresholds. The point cloud data filtered by the intensity threshold will leave some noise points with similar intensity values. The noise points are removed by an outlier removal method. The result is shown in Fig. 3.

3.3　Deriving the transformation by ICP

Let the extracted calibration pattern of the source point cloud be S_{CP1} with m points and the extracted calibration pattern of the target point cloud be S_{CP2} with n points. These two point sets are the input of ICP instead of two whole point clouds.
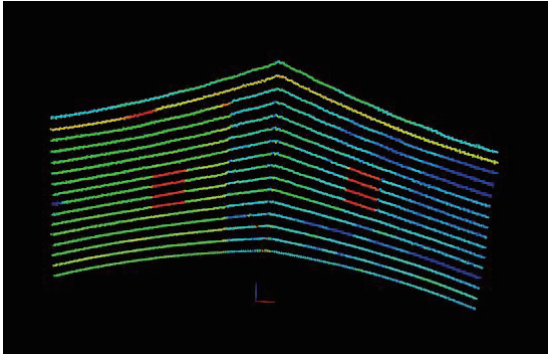
Fig. 2. (Color) The black patterns made by nonwoven fabric have low reflection and the patterns are "highlighted".
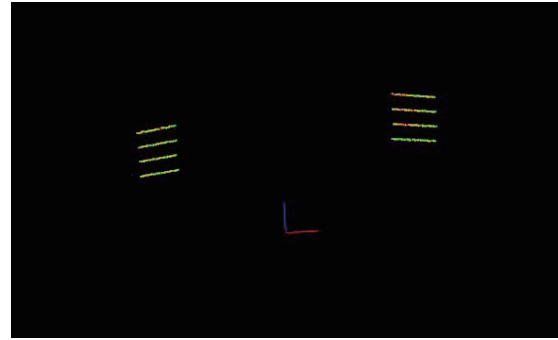

Fig. 3. (Color) Result of calibration patterns filtered by intensity.

After carrying out the procedure introduced in Sect. 2.3, the optimal transformation translation vector T and rotation matrix R can be derived. Trapping at local minima is a big issue in ICP and it happens when the initial point clouds are markedly different. If some known information can be used to roughly adjust the two point clouds first, a better result can be achieved. Therefore, the offset of height can be added to the points of the target point cloud before ICP to obtain a suitable initial position.

3.4 Transforming and extending point clouds

After obtaining the translation vector and rotation matrix between the two point clouds by ICP in the previous subsection, the target point cloud P_T can be transformed into the same coordinate system as the source. The new position of P_T can be calculated by

$$P_T'(i) = \mathbf{R} \cdot P_T(i) + \mathbf{T},$$ (7)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} & \hat{R} & & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$ (8)

where $P_T(i)$ is a point of the target point cloud and $P_T'(i)$ is the new position of the point in the coordinate system of the source point cloud. Then, the extended point cloud is obtained by deleting the overlapping points of the transformed point clouds.

4. Experimental Results and Analysis

The proposed algorithm is simulated by using data captured by a Velodyne VLP-16 LiDAR device mounted on a tripod, as shown in Fig. 4. A Bosch DLE 40 laser rangefinder is also used
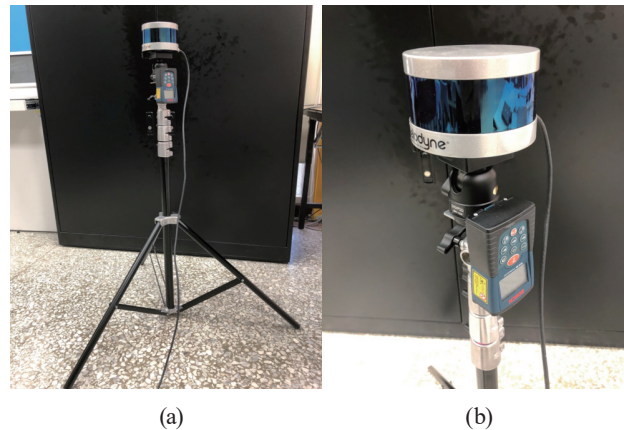
(a)   (b)

Fig. 4.   (Color online) A Velodyne VLP-16 LiDAR device mounted on a tripod (a) and a Bosch DLE 40 laser rangefinder (b).
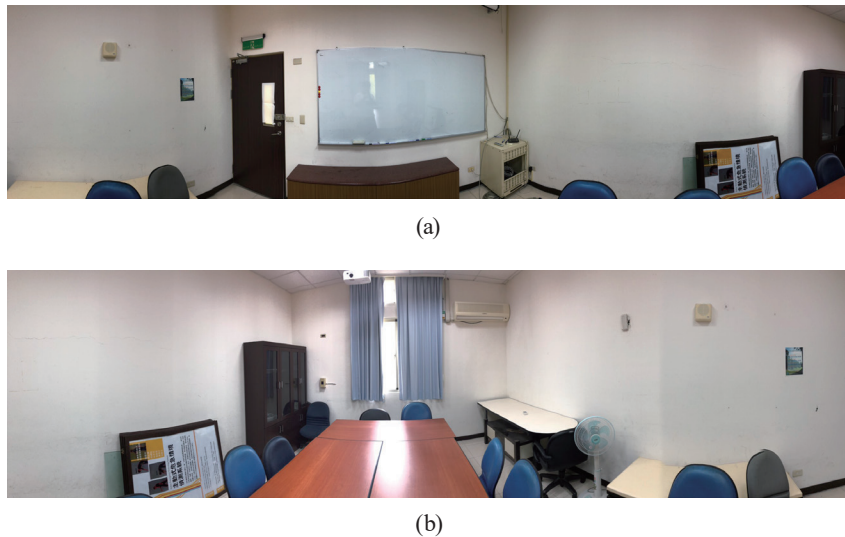

(a)

(b)

Fig. 5.   (Color online) Test environment. (a) Front view. (b) Rear view.

to measure the differences in height. The test environment is a meeting room, which is shown in Figs. 5(a) and 5(b).

In Fig. 6, the points marked in white are the source point cloud and the points marked in red are the target point cloud. To show the true relationship between the two point clouds, the measured height is added to the origin of the target. Thus, the red point cloud is considered as the ground truth. In the first experiment, ICP is used for aligning the two whole point clouds. In Fig. 7, since the ground and the wall dominate the alignment, the extra part of the target is misaligned with the top of the source point cloud. The result of the proposed algorithm using ICP with only the intensity feature block is shown in Fig. 8. The position of the extra part of the target is maintained when aligning. Another simulation was carried out using the pre-known height, which was added as compensation to the coordinates of the points of the target point cloud before ICP. The result is shown in Fig. 9.
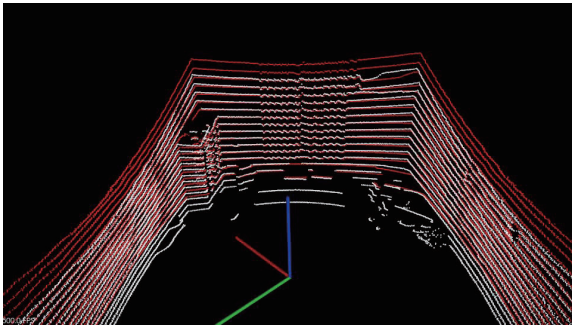
Fig. 6. (Color) Two point clouds are shown in the same coordinate system. The source and the target are in white and red respectively.
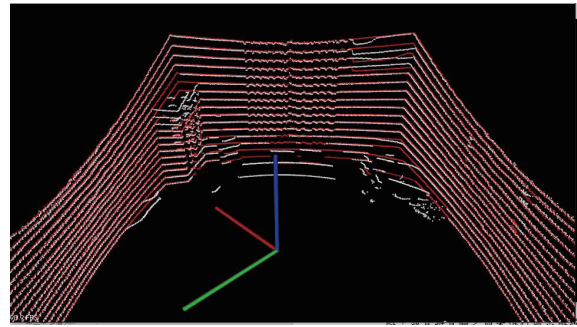

Fig. 7. (Color) Misalignment result of two whole point clouds obtained by ICP. The extra part of the target is misaligned with the top of the source point cloud.
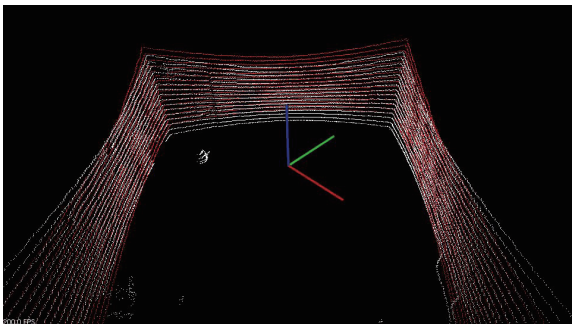

Fig. 8. (Color) Result of the proposed algorithm. The position of the extra part of the target is maintained when aligning.
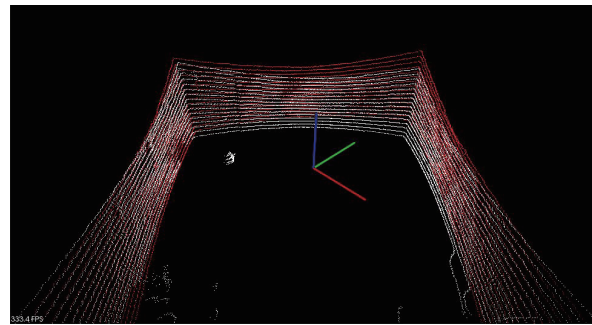

Fig. 9. (Color) Result of the proposed algorithm with a pre-known height. The position of the extra part of the target is maintained when aligning.

5. Conclusions

Since ICP depends on the similarity of the larger parts, e.g., the ground and the wall, the alignment sometimes is incorrect because of the tyranny of the majority. In this paper, an ICP-based registration method only using intensity features is proposed for increasing the vertical FOV of point clouds. The proposed algorithm scans the source and the target with fixed position and varying height, extracts the intensity features from both point clouds, and then derives a better approximate transformation between the two point clouds by ICP.

In simulations, the average error of alignment of the proposed system is less than 16 cm in a 6×6 m^2 room. An improved method using the premeasured height is also tested, where the premeasured height is added to the coordinates of the points of the target point cloud before ICP to improve the performance of the proposed algorithm. The average error of alignment of the proposed system using the premeasured height is less than 12 cm in a 6×6 m^2 room.

Acknowledgments

References

1   O. Cordón, S. Damas, and J. Santamaría: Pattern Recognit. Lett. **27** (2006) 1191. https://doi.org/10.1016/j.patrec.2005.07.017

2   L. Silva, O. R. P. Bellon, and K. L. Boyer: IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 762. https://doi.org/10.1109/TPAMI.2005.108

3   B. K. P. Horn: Proc. the IEEE **72** (1984) 1671. https://doi.org/10.1109/PROC.1984.13073

4   R. J. Campbell and P. J. Flynn: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE 1999) 505.

5   R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin: ACM Trans. Graph. **21** (2002) 807. https://doi.org/10.1145/571647.571648

6   P. J. Besl and N. D. McKay: IEEE Trans. Pattern Anal. Mach. Intell. **14** (1992) 239. https://doi.org/10.1117/12.57955

7   Z. Zhang: Int. J. Comput. Vision **13** (1994) 119. https://doi.org/10.1007/BF01427149

8   A. Gruen and D. Akca: Int'l J. Photo. Remote Sens. **59** (2005) 151. https://doi.org/10.1016/j.isprsjprs.2005.02.006

9   A. W. Fitzgibbon: Image Vision Comp. **21** (2003) 1145. https://doi.org/10.1016/j.imavis.2003.09.004

10   A. D. K. Lam, S. D. Prior, S. Shen, S. Young, and L. Ji: Engineering Innovation and Design (CRC Press, Florida, USA, 2019). https://doi.org/10.1201/9780429019777

11   N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann: Proc. Symp. Geometry Processing (2005).

12   A. D. K. Lam, S. D. Prior, S. Shen, S. Young, and L. Ji: Smart Science, Design & Technology (CRC Press, Florida, 2020). https://doi.org/10.1201/9780429058127

13   A. Makadia, A. Patterson, and K. Daniilidis: Proc. IEEE Comp. Society Conf. Computer Vision and Pattern Recognition (IEEE 2006) 1297–1304.

14   A. E. Johnson and M. Hebert: IEEE Trans. Pattern Anal. Mach. Intell. **21** (1999) 433. https://doi.org/10.1109/34.765655

15   G. C. Sharp, S. W. Lee, and D. K. Wehe: IEEE Pattern Anal. Mach. Intell. **24** (2002) 90. https://doi.org/10.1109/34.982886

16   S. Rusinkiewicz and M. Levoy: Proc. Int. Conf. 3-D Digital Imaging and Modeling (2001) 145–152. https://doi.org/10.1109/IM.2001.924423

17   A. Nuchter, K. Lingemann, and J. Hertzberg: Proc. Int. Conf. 3-D Digital Imaging and Modeling (2007) 419–426. https://doi.org/10.1109/3DIM.2007.15

About the Authors

Chien-Chou Lin received his M.S. and Ph.D. degrees from National Chiao-Tung University, Taiwan, in 1994 and 2004, respectively. From 2010 to 2013, he was an assistant professor, and since 2013, he has been an associate professor at National Yunlin University of Science and Technology, Taiwan. His research interests are robotics, point cloud processing, surface matching, and object recognition. (linchien@yuntech.edu.tw)

Wei-Lung Mao was born in Taiwan, R.O.C., in 1972. He received his B.S. degree in electrical engineering from National Taiwan University of Science and Technology in 1994 and his M.S. and Ph.D. degrees in electrical engineering from National Taiwan University in 1996 and 2004, respectively. He is now a professor in the Department of Electrical Engineering and Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology. His research interests are precision motion control, intelligent and adaptive control systems, satellite navigation systems, adaptive signal processing, neural networks, and communication electronics. (wlmao@yuntech.edu.tw)

Ting-Lun Hu received his B.S. and M.S. degrees in computer science and information engineering from National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2016 and 2018, respectively. Since 2018, he has been a research assistant at National Yunlin University of Science and Technology, Taiwan. His research interests include robotics, path planning, and object recognition. (m10517033@yuntech.edu.tw)