Sensors and Materials, Vol. 36, No. 6 (2024) 2371–2379
MYU Tokyo

S & M 3674

Comparative Analysis of Point Cloud Similarity Based on 3D Surface Reconstruction Using Mechanical Depth Sensor

Wen-Yang Chang,[1,2*] Li-Wei Chen,[1**] Nadia,[1***] and Michael Leandro Hartono[1]

[1]Department of Mechanical and Computer-Aided Engineering, National Formosa University,
No. 64, Wenhua Rd, Huwei Township, Yunlin County 632, Taiwan
[2]Smart Machine and Intelligent Manufacturing Research Center, National Formosa University,
No. 64, Wenhua Rd, Huwei Township, Yunlin County 632, Taiwan

(Received January 3, 2024; accepted June 10, 2024)

Keywords: depth camera sensor, 3D surface reconstruction, point cloud generation, point cloud clustering, passthrough filter, voxel grid, point cloud registration

Currently, significant progress has been achieved in the field of 3D reconstruction, and this trend is anticipated to persist for at least the next decade. In this research, we discuss the utilization of an Azure Kinect depth camera sensor to capture an image of a stamping die and a bottle, recording their depth point values for generating a corresponding point cloud for surface reconstruction. The height of each punch within the die is set manually using a hexagonal key to match the surface of the bottle. The procedural steps involve defining and extracting the field of interest from the original point cloud, incorporating additional filtering, such as passthrough and voxel, to eliminate undesired noise. To enhance processing efficiency during the point cloud registration, clusters are established within the point cloud to distinguish one punch from another and retrieve only the highest point of each. These peak points are then placed within the same coordinate system of the bottle's point cloud for further alignment and obtaining their fitness score at the convergence point. The average discrepancy in dimensions, measured in millimeters, between the actual object and the resulting point cloud is estimated to be less than 10%, with an average time of 3 to 4 min required for the overall surface reconstruction and point cloud registration.

1. Introduction

Depth cameras, also commonly referred to as RGB-D cameras, not only capture regular color images but also retrieve depth data, by means of an infrared sensor, for each pixel at a predetermined configuration.(1–6) This feature makes them highly suitable for performing 3D reconstruction of objects or scenes. An example of such a camera is the Microsoft Azure Kinect DK. While many experiments involve the use of multiple Kinect cameras for a comprehensive model reconstruction, Akay and Akgul noted that this method can be cost-inefficient, and simultaneous communication between multiple cameras may lead to bandwidth issues.(1) Hence, to address this challenge, they adopted a method that used a mirror to generate a virtual point

*Corresponding author: e-mail: wenyang@nfu.edu.tw
**Corresponding author: e-mail: liwei@nfu.edu.tw
***Corresponding author: e-mail: nadiasugianto2000@gmail.com

https://doi.org/10.18494/SAM4886

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

cloud of an object, which was then combined with the nonvirtual or real point cloud to create the overall surface model. Other users make use of an automatic turntable to rotate the object 360° while the camera captures images and retrieves depth data simultaneously at selected angles. Before being fused together at the end, these sets of points may be filtered once or multiple times, such as by using 3D voxel grid filters, to avoid data overlap and long processing time.(2,3) Related articles about point cloud registration methods were investigated for point cloud alignment and reconstruction. (7-10)

2. Methodology

2.1 Point cloud generation

To capture the punch arrangement of the stamping die from the top view, the Azure Kinect depth camera sensor is mounted onto a structure of an aluminum profile and held using screws and bolts, facing downwards, as shown in Fig. 1. Before capturing images, the camera is first set to have the following configurations: 30 frames per second, BGRA 32, 1280 × 720 color image resolution, and wide field-of-view (WFOV) 2 × 2 binned depth mode. Compared with the narrow field-of-view mode, the WFOV mode allows depth data retrieval (in mm) within a field of view of 120° in both directions from a shorter range, starting from 0.25 m. By calibrating the camera using this configuration, a transformation handle is created to transform the coordinate system of the 512 × 512 depth image to that of the 1280×720 color image, while calculating and assigning the real-world *xyz*-coordinate of each pixel within the capture into an empty point cloud.

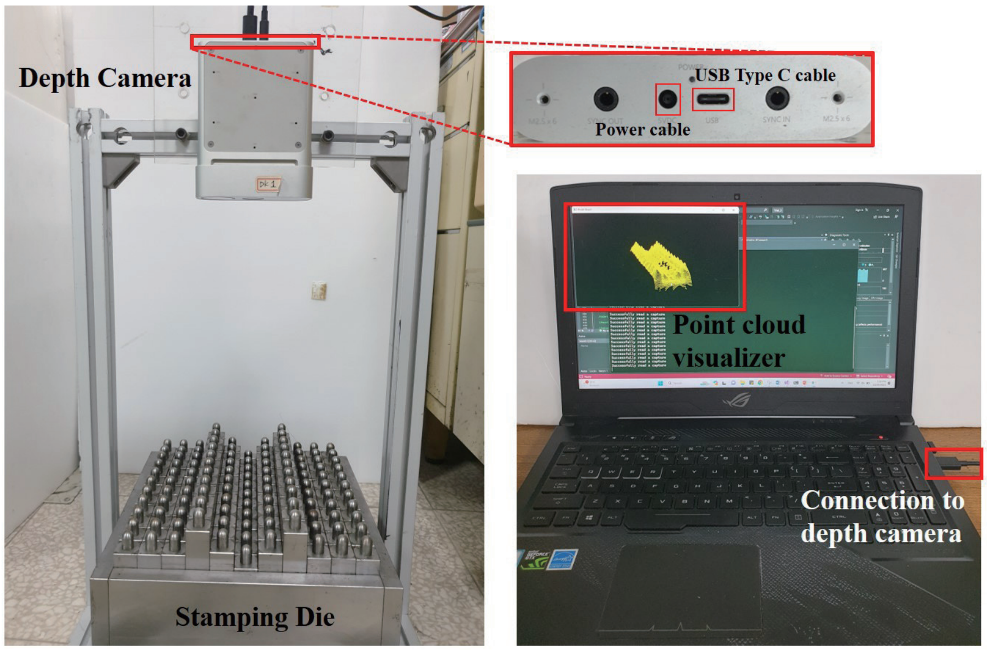


Fig. 1. (Color online) Experimental setup of stamping die, depth sensor, and laptop.

2.2 Point cloud filtering and reconstruction

Each capture of the scene, if unfiltered, has more than one million points, some of which them can be considered as ‘noise’ that is normally removed for reducing processing time. Two methods are included in this point cloud filtering stage, namely, passthrough and 3D voxel grid. The former crops the raw point cloud by declaring maximum and minimum limits of all three axes, focusing the view to the area of interest. Following this, Fig. 2 showcases how the latter then divides the remaining points into grids, in which, to determine the proper size of each grid, several results with different measures are compared, as it is crucial to reduce the point cloud density and reserve the data at the same time. Equations (1)–(6) describe the method to find the minimum and maximum bounds of the entire point cloud based on X_{size}, Y_{size}, and Z_{size} of the grid, to further determine the number of grids (X_{div}, Y_{div}, Z_{div}) required.

$$X_{maxBound} = \frac{X_{max}}{X_{Size}} \text{ and } X_{mixBound} = \frac{X_{mix}}{X_{Size}} \tag{1}$$

$$Y_{maxBound} = \frac{Y_{max}}{Y_{Size}} \text{ and } Y_{mixBound} = \frac{Y_{mix}}{Y_{Size}} \tag{2}$$

$$Z_{maxBound} = \frac{Z_{max}}{Z_{Size}} \text{ and } Z_{mixBound} = \frac{Z_{mix}}{Z_{Size}} \tag{3}$$

$$X_{div} = 1 + X_{maxBound} - X_{mixBound} \tag{4}$$

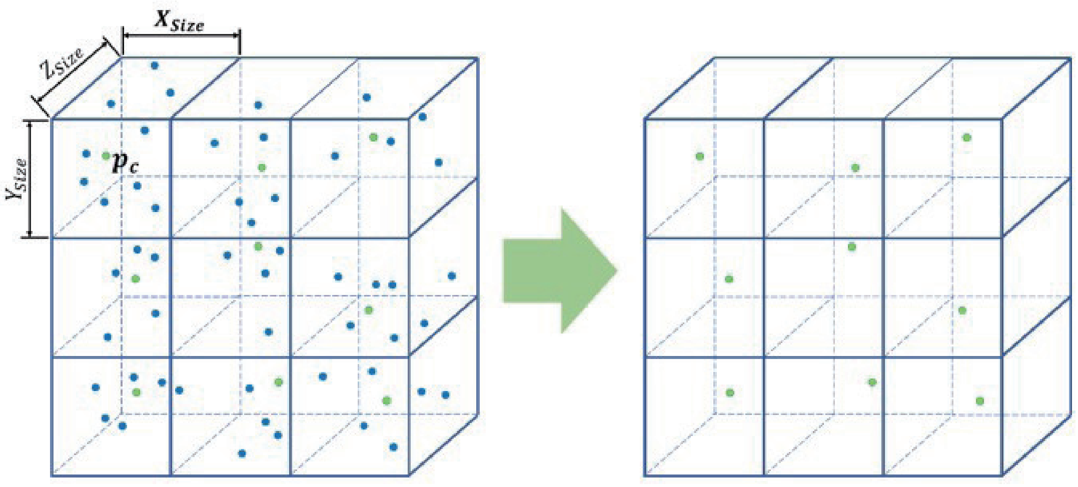


Fig. 2. (Color online) Schematic layout of point cloud voxelization process.

$$Y_{div} = 1 + Y_{maxBound} - Y_{mixBound} \tag{5}$$

$$Z_{div} = 1 + Z_{maxBound} - Z_{mixBound} \tag{6}$$

The *index* value of each point is then calculated using Eqs. (7)–(9), where *i*, *j*, and *k* are the index numbers of the corresponding points in the *x*, *y*, and *z* axes.

$$i = \frac{P_{i_x}}{X_{Size}} - X_{mixBound} \tag{7}$$

$$j = \frac{P_{i_y}}{Y_{Size}} - Y_{mixBound} \tag{8}$$

$$k = \frac{P_{i_z}}{Z_{Size}} - Z_{mixBound} \tag{9}$$

$$index = i + j \times \left(X_{div} \right) + z \times \left(X_{div} \times Y_{div} \right) \tag{10}$$

All points laying within the same grid are then represented only by a new single point, referred to be their centroid, calculated using Eq. (11), where p_c refers to the centroid point and *n* is the number of points within the respective grid.

$$p_c = \frac{1}{n} \sum_{i=1}^{n} p_i \tag{11}$$

To prevent the occlusion of data as a result of downsampling, a moving least squares surface reconstruction method is included to fill in occluded data.(11–12) This method projects existing points within a predefined radius, with respect to their normal, onto a polynomial curve, which aids in smoothing the data as well.

2.3 Point cloud registration

In the case of the stamping die, the crucial part that forms the resulting surface is mainly the tip of the punches.(13–15) To ease the alignment process in the final stage, these ‘peak points’ must be extracted from the point cloud. Therefore, once filtered, by isolating each punch and calculating the Euclidean distance of neighboring points within a predefined radius, as illustrated in Fig. 3(a), clusters are formed to distinguish one punch from another [Fig. 3(b)]. The distance *d* between a point P_i and its surrounding neighboring points P_k can be calculated using Eq. (12).

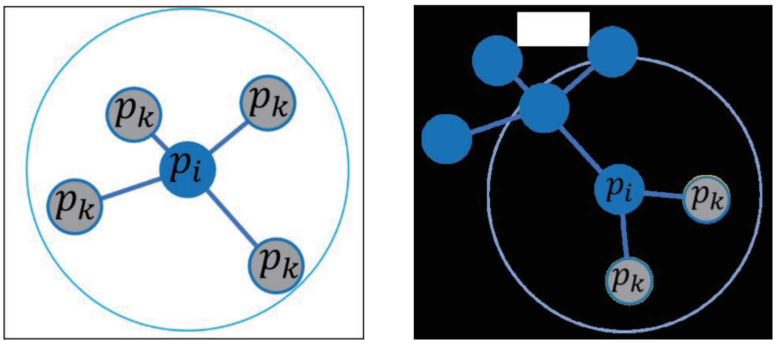


Fig. 3. (Color online) Measuring Euclidean distance (a)between neighboring points within a specified radius and (b) formed to distinguish one punch from another.

$$d(P_i, P_k) = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2} \tag{12}$$

Here, x_i, y_i, and z_i are the x, y, and z coordinates of point P_i and x_k, y_k, and z_k are the x, y, and z coordinates of point P_k, respectively. From each cluster, the point with the lowest z-value (highest from the real-world perspective) is selected and copied into an empty point cloud. Conversely, considering that the point cloud of the bottle is initially inverted upon capture, it necessitates both rotation and additional translation to align accurately with its real-world position for successful alignment.

After both point clouds are acquired and processed, by a point cloud registration method called iterative closest point (ICP), the point cloud containing the peak points is set as the source cloud that is to be aligned to the target cloud, namely, the point cloud of the bottle. Upon reaching the convergence point during the alignment, the fitness score is computed to see if the two point clouds can fit well with each other. A score closer to 0 is preferable as it indicates a minimal error or gap in the difference between the two corresponding point clouds.

3. Results and Discussion

In this study, we focused on a specific area of interest, namely, the central five rows of punches in the die, which were configured to emulate the surface of the bottle. To achieve this, the passthrough filter limits were set as follows: −85 to 27 mm in the x-axis, −150 to 150 mm in the y-axis, and 100 to 350 mm in the z-axis. Determining these values involved conducting multiple iterative trials, applying the limits, and dynamically visualizing the point cloud using the Point Cloud Library Visualizer. Figure 4 shows the resulting point cloud after the passthrough filter has been applied.

Afterwards, the filtered point cloud is divided into grids with size $1 \times 1 \times 1$ mm^3 for its density to be reduced. The difference can be seen between Figs. 5(a) and 5(b), where in Fig. 5(b), there are fewer points within an area, yet the surface structure can still be maintained. The data for every point within a 2 mm radius is then fitted into a 2nd-order polynomial curve. The number of points in the point cloud after each phase can be observed in Table 1.

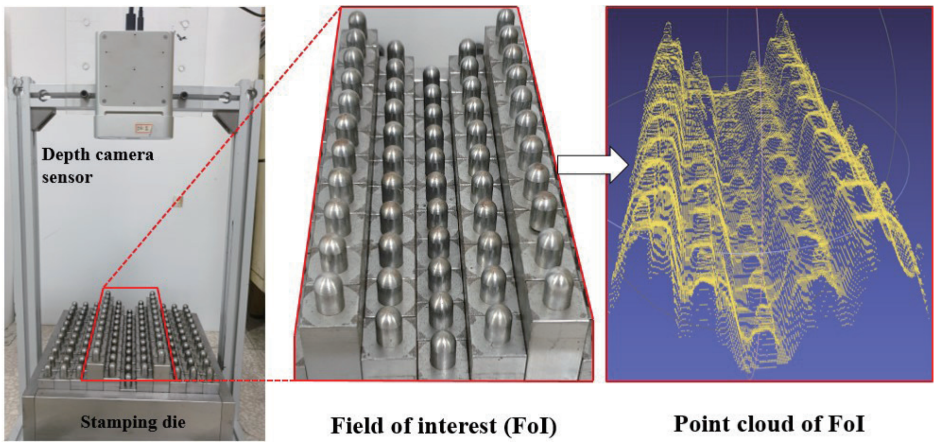


Fig. 4. (Color online) Point cloud of stamping die's field of interest during real-time capture by depth sensor.

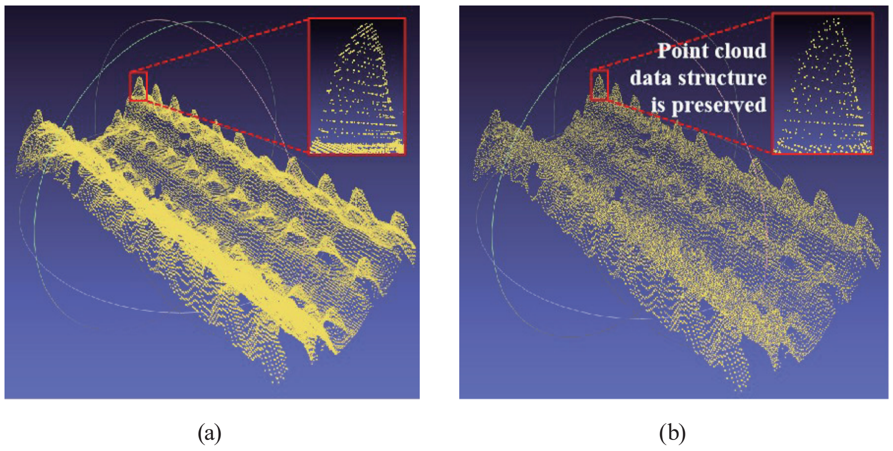


Fig. 5. (Color online) Point cloud of stamping die (a) before voxelization and (b) after voxelization.

Table 1.
Total number of points in stamping die point cloud after downsampling and reconstruction.

Item	Raw	Before voxelization	After voxelization	After polynomial reconstruction
Number of points	1843200	106587	45763	45717

To distinguish each punch in the point cloud, measurements of 23 and 20 mm in both x- and y-axes (estimated on the basis of real distance from one punch to the next) are taken iteratively through the point cloud using a passthrough filter. Gaps of approximately 9 and 5 mm are provided between each cluster in both axes to exclude unnecessary parts. In each iteration, the corresponding isolated points are formed into a single cluster, making a total of 55 main clusters, in which each cluster represents a single punch. The clustering takes 30 to 40 s to form: the overall visualization of the clusters can be seen in Fig. 6(a). From each cluster, the point with the lowest depth value (z-coordinate) is placed into an empty point cloud. This new point cloud thus only consists of points that mark the peak point of each punch. Figure 6(b) shows how the peak points correlate to the original point cloud. Despite having a reduction in the number of points, the peak point positions are not altered. This is an improvement compared with the previous method, which did not result in 100% isolated peaks, as it is fully dependent on the height of the punches. Meanwhile, for the depth sensor to retrieve the point cloud of the bottle successfully, its translucent surface must be covered with paper to prevent the light pass through the surface. Having a symmetrical shape, the point cloud of the bottle is rotated 180° in the y-axis based on the calculated centroid. To bridge the gap from shifting as a result of the rotation, the rotated point cloud must be further translated relative to all three axes. Figure 7 shows a representation of the point cloud during capture, and after rotation and translation.

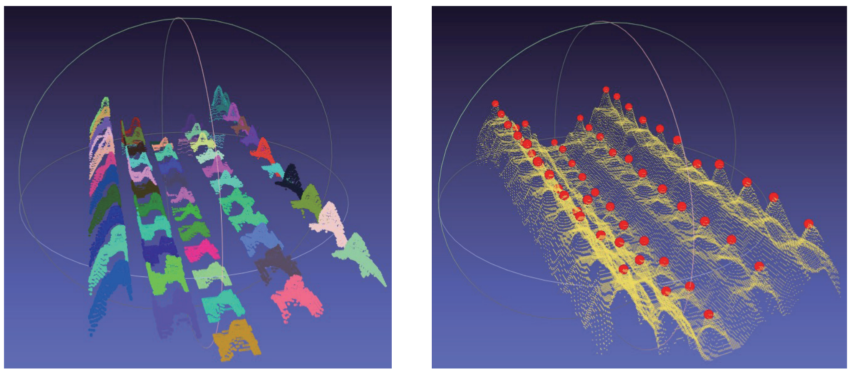


Fig. 6. (Color online) Point cloud of stamping die punch: (a) extraction as a cluster and (b) each cluster corresponding peak point (colored in red).

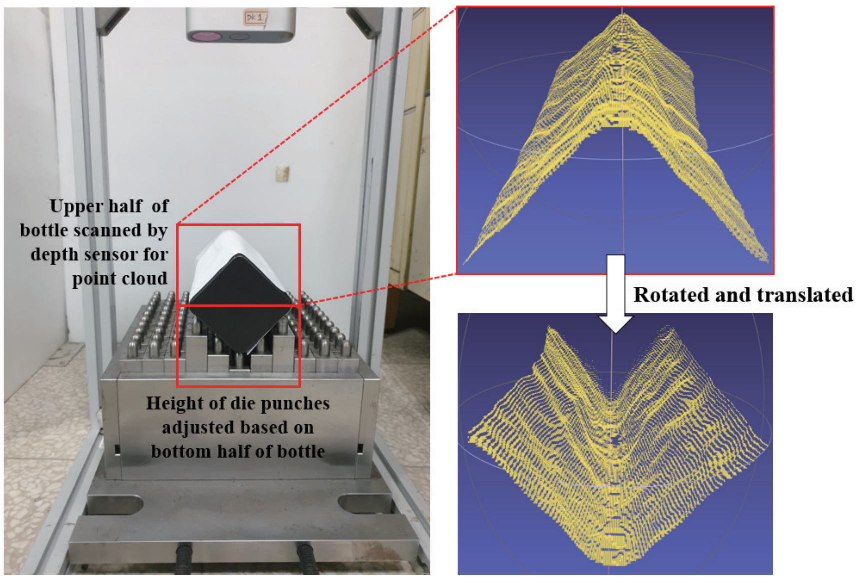

Upper half of bottle scanned by depth sensor for point cloud
Height of die punches adjusted based on bottom half of bottle
Rotated and translated

Fig. 7. (Color online) Point cloud of a symmetrical, square bottle.

In the last stage of the process, employing the ICP algorithm involves designating the peak points as the source cloud, which is then aligned with the target cloud, that is, the point cloud of the bottle. This step, lasting approximately 1 min, yields fitness scores of 8.59823 and 7.29925 at the convergence point when the maximum correspondence distance is set to 10 and 20 mm, respectively. The observed trend suggests that as the tolerance in distance increases, a lower fitness score can be attained. However, experimental results indicate that setting a value higher than 20 does not lead to further reduction of the fitness score. Another result obtained using the original, filtered point cloud, instead of the peak points, leads to a fitness score of more than 100. Therefore, judging by the scores, the alignment between the two point clouds is relatively good, but a more precise adjustment of their coordinate axes is necessary before proceeding with the alignment process.

4. Conclusions

In conclusion, by filtering and downsampling the raw point cloud significantly (more than 90%), the mentioned alignment processes can focus only on the field of interest and thus require less time to finish. For the purpose of this research, determining the length along both axes for each iteration during peak point extraction is an important step as a minor change can result in misplaced peaks. To attain a fitness score better than 7.3, the coordinate axis of the two point clouds should match more accurately. It is recommended to include a trained model for object recognition as it will be more adaptive for other arrangements of punches.

Acknowledgments

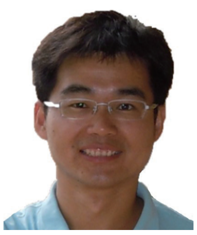
This work was partially supported by the Ministry of Science and Technology, under Grant Nos. NSTC 112-2221-E-150-047 and NSTC 112-2637-E-150-005, and Smart Machine and Intelligent Manufacturing Research Center.

References

1 A. Akay and Y. S. Akgul: Proc. 9th Int. Conf. Computer Vision Theory and Applications (2014). https://doi.org/10.5220/0004676303250334
2 W. Guan, W. Li, and Y. Ren: 2018 Chinese Control and Decision Conf. (CCDC, 2018) 1461. https://doi.org/10.1109/ccdc.2018.8407357
3 X. Teng, G. Zhou, Y. Wu, C. Huang, W. Dong, and S. Xu: Sensors **21** (2021) 4628. https://doi.org/10.3390/s21144628
4 L. Guo, X. Chen, Y. Chen, and B. Liu: Optoelectron. Lett. **11** (2015) 153. https://doi.org/10.1007/s11801-015-5013-2
5 A. Ruchay, K. Dorofeev, and A. Kober: Proc. SPIE 10752, Applications of Digital Image Processing XLI (2018). https://doi.org/10.1117/12.2319911
6 V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche: 2013 IEEE Int. Symp. Mixed and Augmented Reality (ISMAR, 2013) 83–88. https://doi.org/10.1109/ismar.2013.6671767
7 C. R. Popescu and A. Lungu: Computer Science and Information Technology **2** (2014) 95. https://doi.org/10.13189/csit.2014.020206
8 R. Hänsch, T. Weber, and O. Hellwich: ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **II–3** (2014), 57. https://doi.org/10.5194/isprsannals-ii-3-57-2014

9 Z. Zheng, Y. Li, and W. Jun: 2015 IEEE Int. Conf. Progress in Informatics and Computing (PIC, 2015) 588. https://doi.org/10.1109/pic.2015.7489916
10 W. Yookwan, K. Chinnasarn, C. So-In, and P. Horkaew: IEEE Access (2022) 77123. https://doi.org/10.1109/access.2022.3192869
11 S. K. Singh, S. Raval, and B. Banerjee: Int. J. Min. Sci. Technol. **31** (2021) 303. https://doi.org/10.1016/j.ijmst.2021.01.001
12 L. C. Chen, S. H. Huang, and B. H. Huang: Vision Sensors - Recent Advances (Intechopen, Mexico, 2023) 1. https://doi.org/10.5772/intechopen.107968
13 C. L. Kang, T. N. Lu, M. M. Zong, F. Wang, and Y. Cheng: Int. Archives of the Photogramm. Remote Sens. Spatial Inf. Sci. **XLII-3/W10** (2020) 145. https://doi.org/10.5194/isprs-archives-xlii-3-w10-145-2020
14 J. Raschhofer, G. Kerekes, C. Harmening, H. Neuner, and V. Schwieger: Remote Sens. **13** (2021) 3124. https://doi.org/10.3390/rs13163124
15 A. M. Ramiya, R. R. Nidamanuri, and R. Krishnan: Egypt. J. of Remote Sens. Space. Sci. **20** (2017) 71. https://doi.org/10.1016/j.ejrs.2016.04.001

About the Authors



Wen-Yang Chang received his M.S. degree in 2001 from the Department of Mechanical Engineering and his Ph.D. degree in 2008 from the Department of Engineering Science of National Cheng Kung University. He is currently working at National Formosa University. His current research involves the development of smart manufacturing, automatic control and integral systems, and mechanics simulation.

Li-Wei Chen received his M.S. degree from National Taiwan Ocean University, Taiwan, in 2003 and his Ph.D. degree from the University of Sheffield, UK, in 2012. Since 2023, he has been a professor at National Formosa University, Taiwan. His research interests are in advanced metal forming and thermal analysis. (liwei@nfu.edu.tw)

Nadia received her B.S. degree from International University Liaison Indonesia, Indonesia, in 2021. She is currently studying for her M.S. degree at National Formosa University, Taiwan. Her research interests are in 3D surface reconstruction and point cloud processing.

Michael Leandro Hartono received his B.S. degree from International University Liaison Indonesia, Indonesia, in 2021. He is currently studying for his M.S. degree at National Formosa University, Taiwan. His research interests are in system automation, image processing, and object detection.