JAISCR, 2023, Vol. 13, No. 2, pp. 95 – 105
10.2478/jaiscr-2023-0009

A NOVEL METHOD FOR FAST GENERATION OF 3D OBJECTS FROM MULTIPLE DEPTH SENSORS

Tomasz Szmuc[1,*], Rafał Mrówka[1], Marek Brańka[2], Jakub Ficoń[3], Piotr Pięta[1]

[1] *Department of Applied Computer Science, AGH University of Science and Technology,*
al. A. Mickiewicza 30, 30-059 Kraków, Poland

[2] *SynAppsTech, ul. I. Mościckiego 1, 24-110 Puławy, Poland*

[3] *PhD Student AGH University of Science and Technology, Kraków, Poland*
al. A. Mickiewicza 30, 30-059 Kraków, Poland

[*] *E-mail: tsz@agh.edu.pl*

Submitted: 11th November 2022; Accepted: 14th February 2023

Abstract

Scanning real 3D objects face many technical challenges. Stationary solutions allow for accurate scanning. However, they usually require special and expensive equipment. Competitive mobile solutions (handheld scanners, LiDARs on vehicles, etc.) do not allow for an accurate and fast mapping of the surface of the scanned object. The article proposes an end-to-end automated solution that enables the use of widely available mobile and stationary scanners. The related system generates a full 3D model of the object based on multiple depth sensors. For this purpose, the scanned object is marked with markers. Markers type and positions are automatically detected and mapped to a template mesh. The reference template is automatically selected for the scanned object, which is then transformed according to the data from the scanners with non-rigid transformation. The solution allows for the fast scanning of complex and varied size objects, constituting a set of training data for segmentation and classification systems of 3D scenes. The main advantage of the proposed solution is its efficiency, which enables real-time scanning and the ability to generate a mesh with a regular structure. It is critical for training data for machine learning algorithms. The source code is available at https://github.com/SATOffice/improved_scanner3D.

Keywords: 3D scan, LiDAR, point cloud, mesh, registration

1 Introduction

Classification, segmentation, and 3D data generation tasks performed by new algorithms and neural network models achieve statistically significantly worse results than similar tasks for 2D data (images). This fact results from the lack of a spatial relation of any two points to the data position representing those points in the input stream. To overcome this disadvantage, the grid structure and the voxel-based modelling are most often introduced [52]. Unfortunately, the approach introduces other negative consequences: a significant increase in memory consumption by a single 3D scene and the loss of information in the discretization process. Another way to introduce spatial relations is to operate on the so-called regular meshes [49]. In structures of this type, the order of mesh vertices in the input stream for all mesh instances containing the object data is identical. This approach category is usually called the template method. The main advantage is effectively solving the main challenge of the scanning process - shadows and holes. However, they may miss some details of the object itself when they are not reflected in the template.

The paper presents an end-to-end system that generates regular 3D meshes for any objects scanned by devices that read the depth of the image. The method is flexible, allowing for scanning stationary and moving objects with multiple devices. Moreover, the method does not require a manual indicating the correspondence between individual scans, the point cloud, and the template mesh. The presented tool can be used to quickly create new datasets for current and new neural network architectures and artificial intelligence algorithms. The software can be integrated with data sources offered on Kaggle, the open platform that provides Machine Learning and Data Science Community with powerful tools and resources needed for AI experiments [46]. A novelty of the presented work includes the following aspects:

- automatic process of a regular mesh generation based on readings from many synchronized depth sensors that allows performing a scan in less than 500ms for a setup of ten sensors,
- simultaneous use of markers to register multiple point clouds and determine rigid and non-rigid transformations,
- simultaneous use of many types of flat and 3D markers,
- universal system for scanning any 3D objects (shape, size, rigidness) thanks to the use of a dataset of templates.

2 Related works

Acquisition of 3D models basing on the scanning process and geometric modelling or graphical construction of objects is well-known in computer graphics research [1, 5]. Significant technological progress in graphics processors manufacturing and reducing production costs created a space for the dynamic development of the AI / ML area [6, 7, 2]. The progress is also noticeable in scanning and processing 3D images and 3D modelling [3, 4]. However, the demand for new solutions determines a permanent development trend.

2.1 Datasets of 3D models

The basis for the operation of AI/ML algorithms is the availability of appropriate datasets. The experimental activity, algorithm development, and optimization work are impossible without accurate input data. Therefore, the global giants of the IT industry have actively participated in creating and expanding datasets for many areas of life.

Google concentrates on creating high-quality models of common household items [37]. They prepared a dedicated scanning lab for that purpose and introduced many software optimizations. The dataset currently stores more than a thousand scanned objects [38]. ABO Dataset is a result of cooperation between Amazon company and Berkeley University [39]. This dataset consists of listings of products from 576 product types sold by various Amazon-owned stores and websites [40]. Another giant, Facebook AI Research, introduced the Common Objects in 3D (CO3D) [41] dataset comprising 1.5 million multi-view images of almost 19k objects [42]. Finally, another world giant, Alibaba Group company Tao Bao China Software Co [43] introduced the 3D-FUTURE dataset, which provides more than 20 thousand indoor images and the associated unique 3D furniture models with rich geometry details and informative textures[43].

Only several examples were mentioned above. However, the area of creating 3D model datasets is developing fast way. That may be a key driver in fulfilling large-scale datasets designed to help bridge the gap between real and virtual 3D worlds.

2.2 Scanning equipment

There are many scanning methods, as well as many hardware solutions currently available on the market. The progressive miniaturization and continuous expansion of possibilities, which go hand in hand with lowering the equipment cost, significantly increase the opportunities for the entire process and its dissemination. To obtain 3D models, it is necessary to have an expensive studio [29, 30, 31] or expensive mobile scanners [32, 33, 34, 35]. It is possible to use devices such as Intel RealSense cameras, Microsoft Kinect, Intel LiDAR, Structure Sensor, or even smartphone devices [8, 9, 10, 11, 12, 45, 13].

From the methodological point of view, we distinguish depth sensors based on Kinect technology (a grid of infrared beams) and LiDAR sensors. We can find various kinds of sensors available on the market, starting from stationary, manual, and rotary, and ending with smartphone-built. Apart from the size, price, and reliability, each device has its own quality properties that determine the scanning process. The output parameters of the obtained fragments of the scanned surface, configuration options, or the calibration itself are also important here. There are many comparisons of these devices [14, 15]. For the purposes of this study, a further part of the research was based on LiDAR, which looks the most promising.

2.3 Light Detection and Ranging

The method LiDAR (Light Detection and Ranging) relies on measuring distance by illuminating an object with laser light (intense light pulses of a specific wavelength and in a particular direction) and then measuring the light reflected from the object with a sensor. The creation of three-dimensional models (fragments visible by the sensor) is based on the measurement of the difference in the laser beam's return time and the wavelength change. The device itself is a combination of a telescope and a laser sensor with an advanced optical system. The scattering of the light beam resulting from a collision with an object is observed with the device's telescope and then recorded and examined with a glass detector [18, 19].

There are many attempts to apply the solution in various industries. Especially after the manufacturer introduced performance improvements, they are popular in modelling static objects (e.g. buildings or terrain) and in much more ambitious challenges connected with the move and dynamic changes in autonomous driving [17]. However, the use of LiDAR is still limited by the scanning accuracy. So the scanning algorithm and way of processing obtained data still require future optimization [16].

Intel is the market leader in this area and the owner of efficient technology. It is worth emphasizing that the range of RealSense Technology products includes hardware solutions and extended support for developers, including the Intel RealSense SDK 2.0 library [24]. The open-source, cross-platform solution supports the most popular programming languages. It supports depth and colour streaming and advanced calibration information. The library also allows for synthetic streams, including operations on point clouds and depth adjusted to the colour [25]. It is a solution that allows remote management of the device itself [26].

2.4 Not visible parts

The main challenge in creating a complete 3D model based on data from the structured light sensor is missing the object's interior parts and the inclusion of unnecessary background elements. There are many approaches for extracting an object or removing extreme elements of a scene or background [20]. However, they will not be analyzed further due to the concentration of this work on the process of creating 3D objects. It is assumed that the extraction of an object's fragment is carried out correctly. It is achieved by adequately positioning the sensors and arranging the scene to avoid this challenge.

Two main methods to solve the problem of 3D object creation are classical algorithms and machine learning models. Classical ones are based on knowledge about the object through theorems or characteristic properties (axes of symmetry, proportions, etc.). Other methods use the iterative approach to combining point clouds, Simultaneous Localization, and Mapping (SLAM), which combine multiple frames taken over time into the final representation of 3D objects [36].

To carry out the scanning process correctly, the proper arrangement of the devices is crucial. If it is impossible to cover the entire object (because of its size and sensor distance balance), the sensors need to be placed one above the other). It is also essential to put them in such a way that they do not blind each other. For further work, it is assumed that many sensors observe the entire object. Therefore, the challenge will be to faithfully combine these parts to reflect the whole object.

2.5 Point cloud creation

Each device produces its point cloud, representing fragments of the object visible by the device. Of course, they are not perfect. However, the more significant challenge is to synchronize (assemble) them, and remove unnecessary parts, so that you get one collective point cloud that is as close as possible to the real object.

Point cloud manipulation is resource-consuming. Moreover, calculating the shared/common part of the object between two point clouds is even more challenging. It is why in many algorithms, markers are introduced. Usually printed, simple QR codes are used as stickers on the scanned object or in the background [27]. The goal is to locate a marker appropriately, to be seen clearly by at least two sensors.

3 The proposed scanning process

The central assumption of the proposed solution is the possibility of fast, automatic generation of regular 3D meshes using any device that reads the image depth. It also applies to scanning moving objects. Therefore, it became necessary to introduce two steps preceding the generation of the final model in the process. The first step in the process is synchronizing the reading from multiple scanning devices. In the second one, mesh key nodes are recognized thanks to markers placed on the scanned object. Currently, two types of markers are supported: ArUCO (square-shaped, black border with a binary matrix inside) [47] and geometric markers (in our case, coloured spheres).

3.1 Data synchronization from multiple devices

Scanning 3D objects are commonly possible due to the availability of handheld scanners based on LiDAR technology [18, 19] or a structured light [9]. However, the use of this technology is limited to stationary objects only. Objects that cannot remain stationary for a long time, such as items moving on the production lines, cannot be effectively scanned with these technologies. It becomes necessary to run multiple scanners simultaneously.

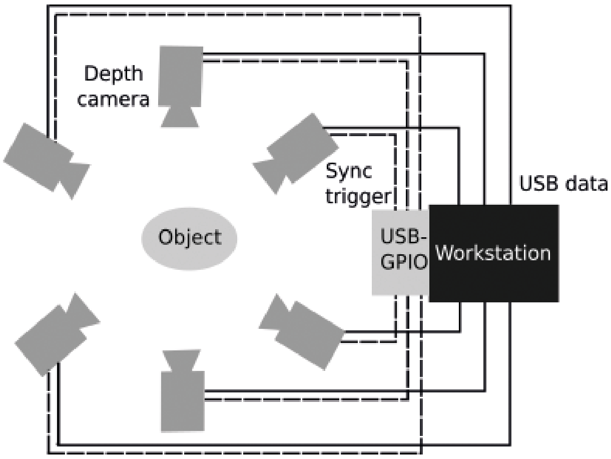

Figure 1. Multiple device object scanning

Unfortunately, this cannot be done due to light interference from multiple independent sources. Therefore, the paper proposes a technology for synchronizing multiple devices, performing a comprehensive scan in no more than 50 ms per device (10ms to warm up the laser and 33ms per frame) [48]. Intel proposed another solution in [48]. However, their solution requires using additional executive units, i.e. RasberryPi.

The proposed solution uses only one computer on which the scanning program has been run with an additional USB-GPIO converter module, for example, FT232H. The scanning program automatically selects which device is to read the object. A scheme of the scanning system implementation is shown in Figure 1, and one of the setups in Figure 2.

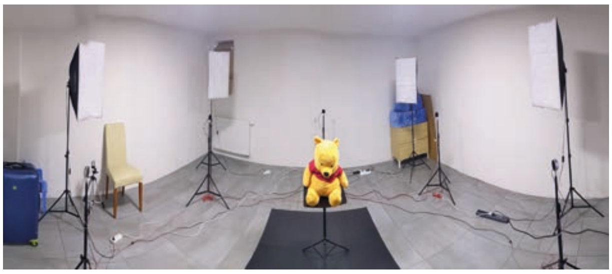

Figure 2. An example for a scene setup

3.2 Identification of key nodes of the mesh

We get many unrelated depth images regardless of the number of data sources (single mobile device, many sequential scanning devices). The registration process of many images into one point cloud is most often carried out using one of the many variants of the ICP algorithm, sometimes supported by data from IMU sensors [22]. For the correct operation of ICP, it is necessary to indicate correspondence points. The most common method is to manually indicate a few correspondence points on the two images we want to merge, generate the remaining points automatically, and then run the ICP algorithm. The result includes a point cloud merged from two or more images. However, the above approach does not solve the correspondence problem with a pattern mesh. It is proposed to use the same correspondence points in the registration process and the non-rigid transformation of the pattern mesh. Thanks to this, the creation of regular 3D meshes is fully automated. The only thing that needs to be done manually is to place the marker of the selected mesh key nodes on the actual object in the appropriate place corresponding to the node location in the mesh. Two types of markers are used in work: flat (ArUCO) and spatial (coloured geometric spheres) [47]. During scanning, the marker type is automatically detected. The information about its identifier and vertex's index of the centre of the marker is saved in the structure describing the point cloud. Radial filtering and downsampling are performed to speed up the calculations. The detected tags are then used to register all depth images using the ICP algorithm.

3.3 Template-scan registration

The mesh template to the point cloud registration process is presented in Figure 3.

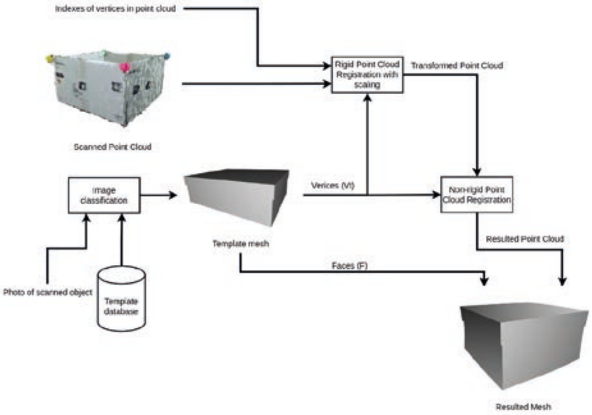

Figure 3. Registration process

After removing the markers from the scanned object, we get a scanned point cloud P_s, which will be used as the target for template-scan registration. P_s can be described as finite set of n_s points:

$$P_s = \{p_i | p_i \in \mathbb{R}^3, i = 1, ..., n_s\} \tag{1}$$

Our source point cloud will be the vertices V_t in the template mesh M_t from the database of template 3D models. The correct template will be selected based on the image classification. Apart from the set of vertices, mesh also consists of a set of faces F_t that will remain unchanged until the end of the algorithm. The lack of faces modification guarantees that the structure will be preserved for all the resulting models using the same template:

$$M_t = \{V_t, F_t\} \tag{2}$$

The V_t set can be described similarly to the P_s but the number of vertices n_t can be different from the number of points n_s:

$$V_t = \{v_j | v_j \in \mathbb{R}^3, j = 1, ..., n_t\} \tag{3}$$

With given two finite size points sets P_s and V_t, our goal is to find such a transformation T^* that will provide the best alignment between the transformed template mesh vertices $T^*(V_t)$ and the scanned model P_s. The best transformation can be described as such transformation T from the set of all transformations τ that the distance function between the transformed pattern and the source point cloud is minimal:

$$T^* = \underset{T \in \tau}{argmin}\, dist(T(V_t), P_s) \tag{4}$$

The distance function between sets of points can be defined as the sum of the distances between each point v and the point closest to it p_m:

$$dist(T(V_t), P_s) = \sum_{v \in T(V_t)} \|t - p_m\| \tag{5}$$

$$p_m = \underset{p \in P_s}{argmin}\, \|t - p\| \tag{6}$$

Thanks to the use of markers on the scanned object, we can split the T^* transformation into two simpler to estimate operations, optimal rigid transformation with scaling T_R^* and optimal non-rigid transformation T_N^*.

For rigid transformation in *dist* function instead of the closest point p_m we will use the set of corresponding points C:

$$C = \{(p_n, v_n)|p_n \in P_s, v_n \in V_t, l = 1, ..., n_c\}. \quad (7)$$

The T_R^* is such a rigid transformation with scaling that the sum of euclidean distances between corresponding points from the set C is minimal:

$$T_R^* = \underset{T_R \in \tau_r}{argmin} \sum_{i=1}^{n} \sqrt{(p_{n1}-v_{n1})^2 + (p_{n2}-v_{n2})^2 + (p_{n3}-v_{n3})^2} \quad (8)$$

This transformation can be defined with three parameters s - scale parameter, R - rotation matrix, and i - translation. With these parameters, we can calculate the transformed position $T_R^*(p)$:

$$T_R^*(p) = sRx + t \quad (9)$$

In non-rigid deformation of the point cloud, we estimate and apply optimal transformation for each individual point p_i and calculate transformation for this point $T_{N_i}^*(p_i)$:

$$T_{N_i}^*(p_i) = s_i R_i x_i + t_i \quad (10)$$

The values of these parameters can be calculated using various algorithms such as Non-rigid Iterative Closest Point (NICP) [21, 22] or Robust point matching [23]. In our implementation, we used the Neural Deformation Pyramid method [21]. We chose this method because it offered very good results for the tested point clouds with a relatively short runtime compared to other tested solutions [50, 51].

The Neural Deformation Pyramid is a hierarchical model in which each level contains a Multi-Layer Perception (MLP). Each perceptron takes as input a sinusoidally encoded 3D point and the output from the previous level. After both registrations, we create a mesh result connecting points in faces as in the original set of faces F_t. The whole process is presented in Figure 4.

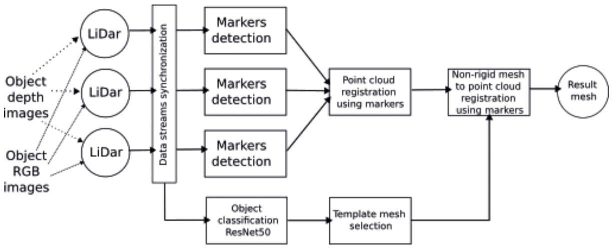
Object
depth
images
Object
RGB
images
LiDar
LiDar
LiDar
Data streams synchronization
Markers detection
Markers detection
Markers detection
Object classification ResNet50
Point cloud registration using markers
Template mesh selection
Non-rigid mesh to point cloud registration using markers
Result mesh

Figure 4. The calculation steps of proposed process

4 Main results

The solution was validated by performing several dozen scans of real objects of various sizes and shapes. Figure 5 presents an exemplary scan of the object. It was assumed that the vertices created as a result of a scan represent the real object points (small fragments). Then, an error function that describes the deformation of a face on the surface was introduced. The error may result from two reasons: a different location of the segment ends and the deformation of the actual surface of the object from the linearly interpolated face in the mesh. Therefore, the mapping error is defined as the sum of the squared interpolation errors for each edge of the mesh:

$$\varepsilon = \frac{\sum_{i \in E} \varepsilon_{i_1} + \varepsilon_{i_2}}{|E|} \quad (11)$$

where E is a set of mesh edges.

The interpolation error is determined as the spatial shift of the edge vertices relative to the actual position corresponding to the vertices of the measurement points on the object surface and the deformation of the interpolated face with respect to the actual plane:

$$\varepsilon_{i_1} = \frac{abs(||p_{i_0}, p_{i_1}|| - ||v_{j_0}, v_{j_1}||)}{||p_{i_0}, p_{i_1}||} \quad (12)$$

v_{j_0}, v_{j_1} - are ends of the edge in the final mesh corresponding to vertices p_{i_0}, p_{i_1},

p_{i_0}, p_{i_1} - are closest points to v_0, v_1 in scan point cloud,

d - is the distance from the centre of the edge to the closes point in the point cloud, and:

$$\varepsilon_{i_2} = \frac{||d||}{||p_{i_0}, p_{i_1}||} \quad (13)$$

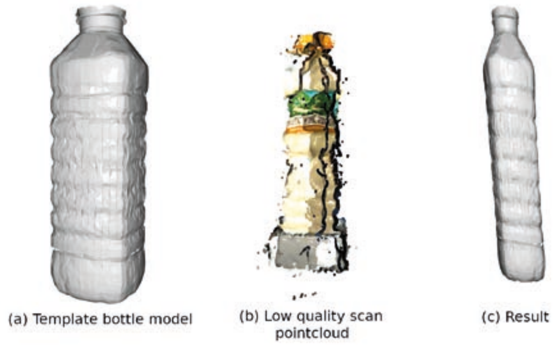

Figure 5. Registration process result: (a) selected mesh template, (b) low-quality scan, (c) final result

The interpolation error of the proposed method for representative scanned objects is presented in Table 1. The choice of objects presented in the table was dictated to determine the dependence of the error value on the scanned object's shape and size. Experimental data show a correlation between the quality of the depth image (depending on many factors, including light conditions) and the interpolation error. Better scan quality translates to lower error. Likewise, the error's value depends on the mesh template quality. A more accurate pattern means less interpolation error. Further work is required to develop the quantitative relation between interpolation error and parameters of the scanning process.

Table 1. Point cloud to average mesh error

Object Category	Models	Mean Error
Box	4	11.8901%
Plastic bottle	5	7.4098%
Ball	3	3.8723%
Tennis racket	2	12.0073%
Toy car	2	12.1287%
Teddy bear	1	15.9376%
Food container	4	8.3023%
Toy snowman	1	6.5412%

5 Conclusion

The lack of publicly available databases of digital models, similar to 2D photo databases, is one of the barriers to developing artificial intelligence algorithms that operate on 3D data. The challenges related to accurately mapping spatial objects to digital models are currently being studied by the most important research institutions worldwide, including large IT companies. Unfortunately, the solutions they use are costly and technically complicated. Moreover, the scanning process is long and cannot scan moving objects. The proposed solution is much easier and more economical to implement, achieving similar accuracy to the generated 3D models.

The proposed method allows for sufficiently fast scanning so that any movements do not affect the quality of the final result. Due to its nature, the technique is ideal for scanning geometrically concave objects. Regardless of the depth sensing device, the surface cannot be scanned in the so-called shadows. The proposed method completes missing information about invisible surfaces based on a mesh template. So we get a detailed model of the object with no holes and a hidden interior. What is more, it does not require complicated mechanical devices. The method is simple and requires only depth sensors, markers, a workstation, and publicly available software. The quality of appropriate mesh templates is the main limiting factor for the proposed method. Nevertheless, the most crucial advantage of the proposed method is that it significantly simplifies the creation of datasets crucial for AI/ML algorithms. That can be especially promising for algorithms operating in real-time on moving objects or living matter.

Acknowledgement

The research project has been supported by the program: Excellence Initiative – Research University for the AGH University of Science and Technology, grant No D4/ID1657: "Development of hybrid classification methods of 3D objects."

References

[1] R. Ramm, M. Heinze, P. Kühmstedt, A. Christoph, S. Heist, G. Notni, Portable solution for high-resolution 3D and colour texture on-site digitization of cultural heritage objects, Journal of Cultural Heritage, vol. 53, pp. 165–175, Elsevier (2022). DOI 10.1016/j.culher.2021.11.006

[2] J. Perez-Cerrolaza, J. Abella, L. Kosmidis, A.J. Calderon, F.J. Cazorla, J.L. Flores, GPU Devices for Safety-Critical Systems: A Survey, ACM Computing Surveys (CSUR), pp.

1–35, New York, NY, USA (2022). DOI 10.1145/3549526

[3] N. Li, C.P. Ho, J. Xue, L.W. Lim, G. Chen, Y.H. Fu, L.Y.T. Lee, A Progress Review on Solid-State LiDAR and Nanophotonics-Based LiDAR Sensors, Laser & Photonics Reviews, vol. 16, 2100511, pp. 1–24, Wiley-VCH GmbH, Weinheim (2022). DOI 10.1002/lpor.202100511

[4] B. Wang, J. Lan, J. Gao, LiDAR Filtering in 3D Object Detection Based on Improved RANSAC, Remote Sensing, Computational Intelligence in Remote Sensing, vol. 14, no. 9, 2110, pp. 1–18 (2022). DOI 10.3390/rs14092110

[5] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, Z. Li, BEVDepth: Acquisition of reliable depth for multi-view 3d object detection, pp. 1–12 (2022). arXiv preprint arXiv:2206.10092

[6] M. Pandey, M. Fernandez, F. Gentile, O. Isayev, A. Tropsha, A.C. Stern, A. Cherkasov, The transformational role of GPU computing and deep learning in drug discovery, Nature Machine Intelligence, vol. 4, no. 3, pp. 211–221 (2022). DOI 10.1038/s42256-022-00463-x

[7] L. You, H. Jiang, J. Hu, C.H. Chang, L. Chen, X. Cui, M. Zhao, GPU-accelerated Faster Mean Shift with euclidean distance metrics, In: IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 211–216 (2022). DOI 10.1109/COMPSAC54236.2022.00037

[8] V. Tadic, A. Toth, Z. Vizvari, M. Klincsik, Z. Sari, P. Sarcevic, J. Sarosi, I. Biro, Perspectives of RealSense and ZED Depth Sensors for Robotic Vision Applications, Machines, vol. 10, no. 3, 183, pp. 1–16 (2022). DOI 10.3390/machines10030183

[9] S. Cerfoglio, C. Ferraris, L. Vismara, G. Amprimo, L. Priano, G. Pettiti, M. Galli, A. Mauro, V. Cimolin, Kinect-Based Assessment of Lower Limbs during Gait in Post-Stroke Hemiplegic Patients: A Narrative Review, Sensors, vol. 22, no. 13, 4910, pp. 1–15 (2022). DOI 10.3390/s22134910

[10] Z. Qiu, J. Martínez-Sánchez, V.M. Brea, P. López, P. Arias, Low-cost mobile mapping system solution for traffic sign segmentation using Azure Kinect, International Journal of Applied Earth Observation and Geoinformation, vol. 112, 102895, pp. 1–11 (2022). DOI 10.1016/j.jag.2022.102895

[11] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, M. Luo, A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LiDAR, Remote Sensing, vol. 14, no. 12, 2835, pp. 1–27 (2022). DOI 10.3390/rs14122835

[12] S. Zahia, B. Garcia-Zapirain, J. Anakabe, J. Ander, O. Jossa Bastidas, A. Loizate Totoricagüena, A Comparative Study between Scanning Devices for 3D Printing of Personalized Ostomy Patches, Sensors, vol. 22, no. 2, 560, pp. 1–20 (2022). DOI 10.3390/s22020560

[13] S. Tavani, A. Billi, A. Corradetti, M. Mercuri, A. Bosman, M. Cuffaro, T. Seers, E. Carminati, Smartphone assisted fieldwork: Towards the digital transition of geoscience fieldwork using LiDAR-equipped iPhones, Earth-Science Reviews, vol. 227, 103969, pp. 1–15 (2022). DOI 10.1016/j.earscirev.2022.103969

[14] P. Chemweno, R.J. Torn, Innovative safety zoning for collaborative robots utilizing Kinect and LiDAR sensory approaches, Procedia CIRP, vol. 106, pp. 209–214 (2022). DOI 10.1016/j.procir.2022.02.180

[15] H.S. Tham, R. Hussin, R.C. Ismail, A Real-Time Distance Prediction via Deep Learning and Microsoft Kinect, In: IOP Conference Series: Earth and Environmental Science, vol. 1064, pp. 1–6 (2022). DOI 10.1088/1755-1315/1064/1/012048

[16] M. Vogt, A. Rips, C. Emmelmann, Comparison of iPad Pro®'s LiDAR and TrueDepth Capabilities with an Industrial 3D Scanning Solution, Technologies, vol. 9, no. 2, 25, pp. 1–13 (2021). DOI 10.3390/technologies9020025

[17] B. Chen, S. Shi, J. Sun, W. Gong, J. Yang, L. Du, K. Guo, B. Wang, B. Chen, Hyperspectral LiDAR point cloud segmentation based on geometric and spectral information, OPTICS EXPRESS, vol. 27, no. 17, pp. 24043–24059 (2019)

[18] T. Staffas, M. Brunzell, S. Gyger, L. Schweickert, S. Steinhauer, V. Zwiller, 3D scanning quantum LiDAR, In: 2022 Conference on Lasers and Electro-Optics (CLEO), pp. 1–2 (2022). DOI 10.1364/CLEO_AT.2022.AM2K.1

[19] F. Di Stefano, S. Chiappini, A. Gorreja, M. Balestra, R. Pierdicca, Mobile 3D scan LiDAR: a literature review, Geomatics, Natural Hazards and Risk, vol. 12, no. 1, pp. 2387–2429, Taylor & Francis (2021). DOI 10.1080/19475705.2021.1964617

[20] A. Notchenko, V. Ishimtsev, A. Artemov, V. Selyutin, E. Bogomolov, E. Burnaev, Scan2Part: Fine-grained and Hierarchical Part-level Understanding of Real-World 3D Scans, In: Proceedings of the 17th International Joint Conference on

Computer Vision, Imaging and Computer Graphics Theory and Applications, vol 5, pp. 711–722 (2022). DOI 10.5220/0010848200003124

[21] Y. Li, T. Harada, Non-rigid Point Cloud Registration with Neural Deformation Pyramid, pp. 1–19 (2022). arXiv preprint arXiv:2205.12796

[22] P. Besl, H.D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 14, pp. 239–256 (1992). DOI 10.1109/34.121791

[23] S. Gold, A. Rangarajan, C. Lu, S. Pappu, E. Mjolsness, New algorithms for 2D and 3D point matching: pose estimation and correspondence, Pattern Recognition, vol. 31, no. 8, pp. 1019–1031 (1998). DOI 10.1016/S0031-3203(98)80010-1.

[24] Intel Corporation. Intel RealSense Product Overview. Retrieved October 30, 2022, from https://www.intelrealsense.com/

[25] Intel Corporation. Learn About the Intel® RealSense^{TM} SDK 2.0. Retrieved October 30, 2022, from https://www.intelrealsense.com/intel-realsense- sdk-2-0/

[26] Intel Corporation. Intel® RealSense^{TM} LiDAR Camera L515. Retrieved October 30, 2022, from https://www.intelrealsense.com/LiDAR-camera-l515/

[27] Intel Corporation. Multi-Camera configurations with the Intel® RealSense^{TM} LiDAR Camera L515. Retrieved October 30, 2022, from https://dev.intelrealsense.com/docs/LiDAR-camera- l515-multi-camera-setup

[28] H. Sarmadi, R. Muñoz-Salinas, M.A. Berbís, A. Luna, R. Medina-Carnicer, Joint scene and object tracking for cost-effective augmented reality assisted patient positioning in radiation therapy, pp. 1–16 (2020). arXiv preprint arXiv:2010.01895.

[29] The Gnomon Workshop. 3D scan and retopology for production. Retrieved October 30, 2022, from https://www.thegnomonworkshop.com/tutorials/3d-scan-and-retopology-for-production

[30] ScanLab photogrammetry. 3D scanning service. Retrieved October 30, 2022, from https://scanlab.ca/services/3d-scanning/

[31] Scan Engine. Studio. Retrieved October 30, 2022, from https://www.scan-engine.fr/

[32] Shining 3D. 3D Digitizing Solutions. Product models. Retrieved October 30, 2022, from https://www.shining3d.com/3d-digitizing-solutions/

[33] Artec 3D. Artec Leo 3D scanner. Retrieved October 30, 2022, from https://www.artec3d.com/portable-3d-scanners/artec-leo

[34] Peel 3D. Peel 3 3D scanner. Retrieved October 30, 2022, from https://peel-3d.com/products/peel-3/

[35] Calibry. Skaner 3D Calibry. Retrieved October 30, 2022, from https://calibry.pl/

[36] A. Kulikajevas, Reconstruction algorithm of invisible sides of a 3D object for depth scanning systems, Doctoral dissertation (2022), Kauno technologijos Universitetas, KTU, Lithuania

[37] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T.B. McHugh, V. Vanhoucke, Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items, pp. 1–8 (2022). arXiv preprint arXiv:2204.11918

[38] Google Research. Scanned Objects by Google Research: A Dataset of 3D-Scanned Common Household Items. Retrieved October 31, 2022, from https://ai.googleblog.com/2022/06/scanned-objects-by-google-research.html

[39] Amazon. Amazon Berkeley Objects (ABO) Dataset. Retrieved October 31, 2022, from https://amazon-berkeley-objects.s3.amazonaws.com/index.html#home

[40] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, T.F.Y. Vicente, T. Dideriksen, H. Arora, M. Guillaumin, J. Malik, ABO: Dataset and benchmarks for real-world 3d object understanding, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21126-21136 (2021). DOI 10.1109/CVPR52688.2022.02045

[41] Meta. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. Retrieved October 31, 2022, from https://ai.facebook.com/datasets/CO3D-dataset/

[42] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, D. Novotny, Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction, In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10901–10911 (2021). DOI 10.1109/ICCV48922.2021.01072

[43] Alibaba. Our Businesses. Retrieved October 31, 2022, from https://www.alibabagroup.com/en-US/about- alibaba-businesses

[44] H. Fu, R. Jia, L. Gao, M. Gong, B. Zhao, S. Maybank, D. Tao, 3d-future: 3D furniture shape

with texture, International Journal of Computer Vision, vol. 129, no. 12, pp. 3313–3337 (2021). DOI 10.1007/s11263-021-01534-z

[45] XRPro LLC (Structure). Structure Sensor Pro. Retrieved October 31, 2022, from https://structure.io/

[46] Kaggle Inc. Datasets. Retrieved October 31, 2022, from https://www.kaggle.com/datasets

[47] F.J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, Speeded up detection of squared fiducial markers, Image and Vision Computing, 76, pp. 38—47 (2018). DOI 10.1016/j.imavis.2018.05.004

[48] Multi-Camera configurations with the Intel® RealSense^{TM} LiDAR Camera L515. Retrieved October 31, 2022 from https://dev.intelrealsense.com/docs/LiDAR-camera-l515-multi-camera-setup

[49] V. Kraevoy, A. Sheffer, Template-Based Mesh Completion, Eurographics Symposium on Geometry Processing (2005). DOI 10.2312/SGP/SGP05/013-022

[50] J. Huang, T. Birdal, Z. Gojcic, L. Guibas, S. Hu, Multiway Non-rigid Point Cloud Registration via Learned Functional Map Synchronization, In: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–18 (2022). DOI 10.1109/TPAMI.2022.3164653

[51] Y. Li, T. Harada, Lepard: Learning partial point cloud matching in rigid and deformable scenes, pp. 1–17 (2021). DOI 10.48550/ARXIV.2111.12591

[52] F. Poux, R. Billen, Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featuring vs. deep learning methods, ISPRS International Journal of Geo-Information, vol. 8, no. 5, 213, pp. 1–34 (2019). DOI 10.3390/ijgi8050213

Tomasz Szmuc received M.Sc. (1972) in Electrical and Control Engineering from the AGH University of Science and Technology (AGH). Since the beginning, he has been employed at AGH University of Science and Technology, where he received Ph.D. (1979) and Sc.D. (1989) degrees (both in Computer Science), and Professor title (1999). His research may be assigned to two main areas: formal methods supporting software development and hybrid approaches in big data analysis. The first area focuses on applications of Petri nets, process algebras, and temporal logics in the development. The combined use of Fuzzy-Rough Sets and neural networks forms the basis of his second research branch. https://orcid.org/0000-0003-4922-5369

Rafał Mrówka was employed as an Assistant at AGH University of Science and Technology in 2007. In 2008 he defended his doctoral dissertation regarding formal methods in software engineering and began working as a lecturer. During this period, he conducted research and teaching activities requiring practical knowledge of Software Engineering. He co-creators the Alvis formal modelling language based on process algebra. He published several articles dealing with the subject of UML translation into Petri nets and the use of bisimulation in the process of verifying the requirements of real-time systems. Recent research has focused on using artificial intelligence in practical software engineering. In particular, he is interested in automated processing and generating spatial models of objects using artificial neural networks. https://orcid.org/0000-0001-5143-3488

Marek Brańka received MSc in Telecommunication from the AGH University of Science and Technology in Cracow, Poland. He pursues his research interests as part of the "Smart methods in software engineering and data analysis" research group, operating at the Department of Applied Computer Science of the AGH University of Science and Technology. His primary research focuses on 3D scanning systems and integrating the latest achievements in computer science, including artificial neural networks and intelligent embedded systems with software engineering. He serves as a CTO of SynAppsTech, an international software house, and supports various R\&D projects and end-to-end processes of high-tech product delivery. Experienced IT leader, serving as the head of IT departments providing services for financial, e-commerce, and automotive markets. He is a new technology enthusiast, providing his consultancy services for startup environments and participating in hackathons and conferences as a mentor and expert. https://orcid.org/0000-0002-4732-2176

Jakub Ficoń graduated in 2022 with Computer Science and Intelligent Systems degree. During his studies, he worked on projects related to applications of VR technology for education and the use of Reinforcement Learning in computer games. Also, in 2022, he began his studies at the AGH Doctoral School. While still in his Master's Degree Program, he participated in the assistantship program at the Department of Applied Computer Science, where he co-taught and collaborated on research related to Software Engineering. Since 2022, he has worked in the Operations Research team at CAE.
https://orcid.org/0000-0001-8703-1413

Piotr Pięta is a Ph.D. student at the AGH University of Science and Technology in Cracow, Poland. He received MSc in Computer Science in 2018. P. Pięta is a member of "Smart methods in software engineering and data analysis" research group at the Department of Computer Science AGH. His research activity over the last years focused on various aspects of the Rough Sets Theory in big data analysis and data mining, Fuzzy-Rough Sets, Machine Learning, Collective Intelligence, and Artificial Intelligence.
https://orcid.org/0000-0001-8490-7117