INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.232

Towards digital representations for brownfield factories using synthetic data generation and 3D object detection

Javier Villena Toro ✉, Lars Bolin, Jacob Eriksson and Anton Wiberg
Linköping University, Sweden
✉ javvi51@liu.se

Abstract
This study emphasizes the importance of automatic synthetic data generation in data-driven applications, especially in the development of a 3D computer vision system for engineering contexts such as brownfield factory projects, where no data is readily available. Key points: (1) A successful integration of a synthetic data generator with the S3DIS dataset, leading to a significant enhancement in object detection of previous classes and enabling recognition of new ones; (2) A proposal for a CAD-based configurator for efficient and customizable scene reconstruction from LiDAR scanner point clouds.

Keywords: digital twin, brown field, synthetic data generation, point cloud, artificial intelligence (AI)

1. Introduction

In the pursuit of automation and the integration of new technologies, greenfield factory projects enjoy a distinct advantage as they can seamlessly incorporate the latest technologies and considerations right from the initial design phase. One notable and extensively researched topic in this context is the introduction of a digital twin, which serves as a model representation that seamlessly integrates with the factory, facilitating two-way communication between the physical entity and its digital counterpart (Piascik R., 2010). In contrast, brownfield factories with pre-existing structures and legacy software systems may not be the ideal candidates for implementing a digital twin (Sierla et al., 2020).
As a viable solution, some authors have suggested the concept of "experimentable digital twins" (Schluse *et al.*, 2018). This approach to digital twins leverages existing 3D models of the plant, and retrofitting is achieved by strategically placing sensors throughout the factory (Martínez *et al.*, 2018). These "experimentable digital twins" offer essential benefits akin to full operational digital twins, including the capability for steady-state and dynamic simulations (Sierla et al., 2020).
In many cases, an updated 3D model of the brownfield factory may not be readily available (Chen *et al.*, 2018). This can happen either because there was never a model in the first place or because subsequent structural and equipment changes have rendered the existing model obsolete. In this challenging scenario, engineers face two potential alternatives:
The first option is to manually reconstruct the factory's model, which can be a labor-intensive process demanding a significant amount of engineering effort (Sierla, Azangoo, *et al.*, 2020). The second alternative involves creating a non-functional geometry using a 3D scanner to generate a point cloud representation of the factory. Modern technology, such as the LiDAR scanner found in smartphones and tablets, can provide reliable accuracy in capturing the macroscopic environment (Wang *et al.*, 2020). However, it's important to note that this simplified geometry would limit the capabilities of the digital twin, primarily allowing for use cases like collision analysis (Shellshear *et al.*, 2015).

Recent breakthroughs in 3D computer vision have opened exciting possibilities for employing neural networks in object detection and segmentation from point clouds (Qi, Su, *et al.*, 2017). These advancements have the potential to revolutionize the automation of 3D models for brownfield plants using 3D scanners. There is still potential for improving the accuracy of predictions, and the number of classes available in pretrained models for 3D data is currently less than those for 2D data (Chen *et al.*, 2023). However, it is possible to develop specialized frameworks for training neural networks that are specifically designed to address the unique requirements and challenges of engineering applications in a 3D context. This presents a promising avenue for further innovation and automation in the field of brownfield plant modelling.
This paper endeavors to address the existing gap in the utilization of high-end point cloud detectors within engineering contexts. The paper offers a threefold contribution. First, we introduce a synthetic data generator designed for indoor point clouds. Second, we present a deep learning approach that leverages both synthetic generated data and existing datasets, along with an analysis of the performance. Additionally, we introduce a configurator that facilitates the conversion of the neural network output into a customizable computer-aided design (CAD) environment.
The paper's structure is as follows. In Chapter 2, we introduce deep learning in the context of 3D environments, as well as an exploration of various approaches to synthetic data generation. Chapter 3 presents the developed framework divided into three sections. The first section focuses on synthetic data generation; the second on model training and evaluation; and the third section elaborates on the configurator. Chapter 4 offers a comprehensive presentation of the insights and reflections derived from the results obtained through our research. Lastly, in Chapter 5, we present the conclusions drawn from our work.

2. Background study

2.1. Computer vision for point clouds

The field of computer vision is often referred to as a domain that empowers computers to extract meaningful information from visual input. While traditional computer vision algorithms excel at solving classification, object detection, and segmentation problems in images and video recordings, the surge in available 3D data and the development of algorithms tailored for such formats have propelled computer vision into the 3D realm.
Various strategies have been employed to address the challenge of increased dimensionality. Multiview Convolutional Neural Networks (CNNs) project 3D objects or scenes into 2D images, primarily for classification purposes (Su *et al.*, 2015). Feature-based Deep Neural Networks (Fang *et al.*, 2015) convert 3D inputs into feature vectors to extract information, while Spherical CNN (Esteves *et al.*, 2017) projects data onto a sphere and processes it using spherical filters. Additionally, Volumetric CNNs (Maturana and Scherer, 2015) apply 3D CNNs to voxelized shapes.
PointNet introduced a paradigm shift in 3D deep learning by pioneering a network architecture that operates directly on point cloud data, dispensing with the reliance on predefined grids or structures. Its key innovation lies in the symmetric function design, ensuring permutation invariance and enabling the network to process unstructured point sets (Qi, Su, *et al.*, 2017). Subsequent developments, such as PointNet++ (Qi, Yi, *et al.*, 2017) and VoteNet (Qi *et al.*, 2019), have furthered this progress. VoteNet, for instance, employs a voting mechanism to iteratively refine object proposals, enhancing robustness and accuracy in identifying objects within complex 3D scenes. These advancements, including one of the latest releases, TR3D (Rukhovich *et al.*, 2023), which has achieved state-of-the-art performance (as of June 2023) with improved accuracy and speed, can be trained, developed, and benchmarked using the openMMlab platform (MMDetection3D, 2020).
After the background search for the most suitable algorithm to process the point clouds, the TR3D architecture was chosen for the deep learning component of the framework.

2.2. Data available and synthetic data generation

Synthetic data refers to information that, although not obtained through the measurement of physically existing objects, closely resembles real data (Anderson *et al.*, 2014). The generation of synthetic data is gaining increased traction within the deep learning community, particularly as the primary bottleneck for their architectures arises from the inadequacy of sufficiently large datasets (Nguyen *et al.*, 2022). This shortage of data becomes even more conspicuous in fields like design and manufacturing, where the demand for highly specific data exceeds what is publicly available.

Approaches to synthetic data generation for point cloud data can be broadly categorized as simple or advanced (Nguyen *et al.*, 2022). Simple methods involve the conversion and annotation of CAD models into point clouds, primarily tailored for classification problems. Examples of such synthetic data generation include ModelNet40 (Wu *et al.*, 2015) and ShapeNet (Chang *et al.*, 2015). On the other hand, advanced techniques employ automatic scene generator frameworks meticulously crafted to generate scenes and record annotations with high precision (Nguyen *et al.*, 2022). This paper introduces an automatic synthetic data generator aligned with these advanced methodologies.

While instances like the Biked dataset (Regenwetter *et al.*, 2021) demonstrate scenarios where synthetic data and models are co-designed, in computer vision, developers often opt for transfer learning strategies. In such cases, synthetic data needs adaptation to align with the format and annotation system of existing data and model architecture. For point cloud datasets, the absence of a standard labeling method necessitates synthetic data to adhere to one of the proposed annotation systems. Notably, popular open-source datasets for interior point clouds (or RGB-D) include ScanNet (Dai *et al.*, 2017), Sun RGB-D (Song *et al.*, 2015), and S3DIS (Armeni *et al.*, 2017).

After evaluating these three popular datasets, the choice was made to select S3DIS over the other two options. The reason for choosing it over Sun RGB-D is that the latter is constructed based on 2D images with depth pixel information, rather than utilizing raw point cloud data. Additionally, S3DIS was favored over ScanNet due to its more straightforward annotation system, which enhances the efficiency of the data generation process.

3. Framework

The objective of this paper is to create a data processing pipeline that automates the reconstruction of factory environments. The framework we present bears a resemblance to the one proposed by Nguyen et al. (Nguyen *et al.*, 2022). Figure 1 provides a flowchart illustrating the various stages discussed in this chapter, with distinct sections clearly demarcated using colors. Before commencing the automatic synthetic data generation process, two crucial decisions need to be made. These decisions have been already discussed in the background study.

3.1. Synthetic data generation

Common 3D datasets, like the S3DIS database, typically do not account for the specialized equipment commonly found in manufacturing or process plants. The S3DIS dataset classes mainly consist of structural elements (e.g., ceiling, floor, wall, beam, column, door, window) and movable elements (such as tables, chairs, sofas, bookcases, boards), along with a separate category for "clutter," encompassing the remaining points in the scene.

While structural elements are crucial for segmentation tasks, there are alternative methods to determine the scene's size (as outlined in Section 3.3) that can more effectively allocate model resources towards the detection of movable objects. Consequently, certain authors, like the developers of TR3D, choose to evaluate their model against only the 5 movable objects (Rukhovich *et al.*, 2023) to better address the specific needs of their application.

To complement the S3DIS dataset, we introduced additional classes in the generated data scenes. Complementing classes include "table," "chair," and "sofa," which are already present in the S3DIS dataset. The two added classes are:

- The ABB collaborative robot YuMI (IRB-14000) (ABB, 2015), labeled as "yumi." This class represents a parametric model, with the arm joint positions as parameters.
- A highly geometrically variable product, such as 3D printers, labeled as "3dprinter."

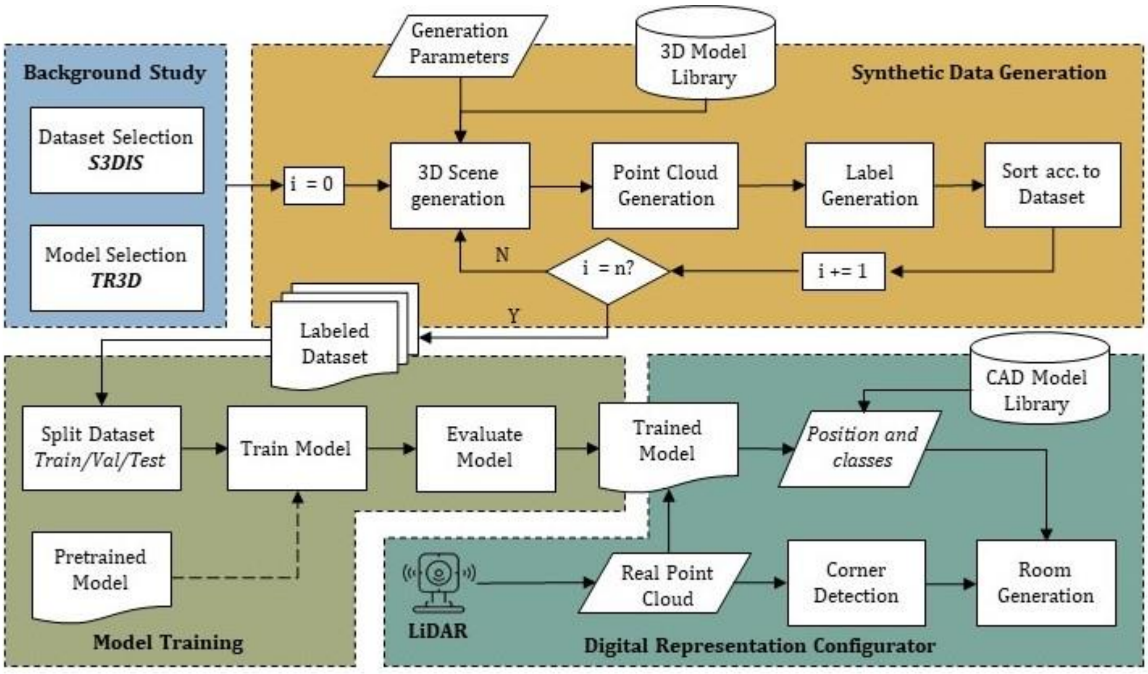

Figure 1. Framework for achieving digital representations based on object detection boosted with synthetic data generation. "n" is the number of desired data samples

The selection of these classes serves several purposes, with the goal of showcasing the advantages of using synthetic data:

a) Demonstrating the TR3D detection algorithm's ability to learn distinctive features associated with the newly added classes.
b) Providing evidence that synthetic datasets can enhance the performance metrics for existing classes (i.e., "table," "chair," "sofa") or, at the very least, not degrade their performance when new instances are included.
c) Assessing the model's capacity to detect the newly added classes in real point clouds, thereby demonstrating its generalization capability.

The 3D software employed for scene generation is Blender, a robust open-source 3D tool that offers an open Python application programming interface (API) for customization. At a higher level, the generation framework operates through a loop, receiving general parameters to control various aspects, including the number of scenes to be generated, point number and density (density refers to the distribution of points between movable objects and the background elements).

The model collection comprises five Blender files, one for each class, and within these files, one can find 6-15 different instances of each class. For common movable objects like chairs, tables, and sofas, the source files were obtained from open-source Blender files available on BlenderKit. Instances of the YuMI robot were created after manually manipulating the joints within commercial CAD software, and 3D printer models were collected in "step" files from various sources such as GrabCAD or Printables webpages. These files were then converted into mesh format for inclusion in the model collection.

To ensure the generation of scenes that avoid object intersections and appear somewhat realistic (as depicted in Figure 2a), the following rules have been implemented:

- Room structural elements are represented by a parametric cube that encapsulates all elements within the room.
- Each object possesses a parameter governing its appearance probability. Consequently, while empty scenes are possible, they are highly improbable.
- Tables and sofas are positioned along the walls, sharing this space. As a result, the maximum number of sofas or tables per scene is limited to four.
- Chairs are placed within an imaginary inner boundary of the room at four specific locations. They have the flexibility to be rotated.

- 3D printers and YuMi robots can only be placed on top of tables and share the available space. Each table can accommodate two elements, one on each side (local left and right).

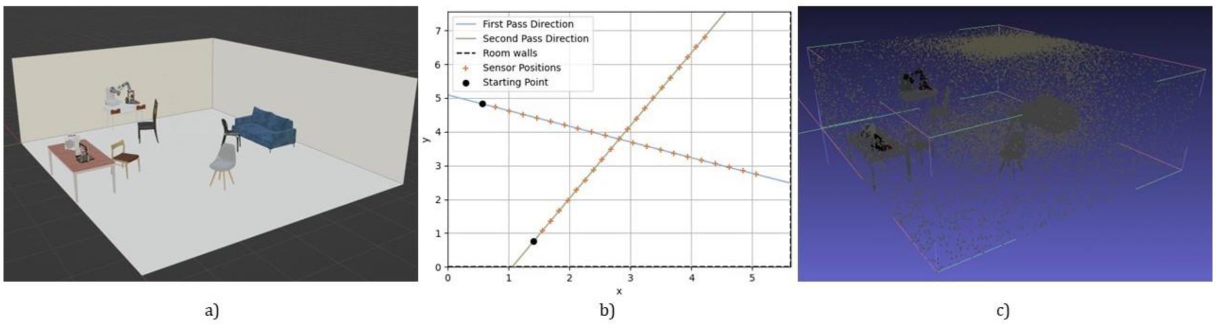

Figure 2. a) Sample scene generation; b) Sensor paths and ray cast locations; c) Generated point cloud; Note that the colors of textured objects cannot be captured

After the scene is generated, the point cloud generation process is initiated, drawing inspiration from Helios++ (Winiwarter *et al.*, 2021), but adapted for indoor use cases. Helios++ is a software designed for simulating laser scanning in both static and dynamic outdoor environments. In this article, a pivotal role is played by a Blender function known as "Raycast." This function projects a ray from an emissary object, referred to as the sensor, and determines whether there was a collision or not with a target object. It also provides information about the local coordinates of the hit, surface normal, and the face ID. The function takes as input the sensor object, the target object, and the ray's direction (Blender F., n.d.).
Like the Helios implementation, the sensor follows a predefined path, emitting rays at each step. This path is divided into two segments that intersect covering only 80% of the room's area, as depicted in Figure 2b. The height of the sensor is equivalent to 2/3 of the room's height. The direction of the path is determined by a random angle. The ray directions are entirely randomized, and to ensure a representative number of points on movable objects, the "room object" is excluded from the list of possible targets. Once the desired proportion of points has been reached, rays are projected into the room's structural elements.
Each local coordinate and face information is recorded to facilitate the retrieval of global coordinates and, when possible, the material color of the face. This process results in a matrix consisting of N lines, representing the total number of points. Each line contains the following point information: [x, y, z, r, g, b, c], where the first three elements represent the point's coordinates, followed by the RGB color information and the object class. The point cloud data is stored in separate files and folders, following the same fashion as the S3DIS dataset. The point cloud is illustrated in Figure 2c.
For the experiment detailed in this article, the generated data includes three new areas, each containing 50 scenes. The representation of movable objects in the combined dataset (comprising both S3DIS and synthetic data) is summarized in Table 1.

Table 1. S3DIS dataset and generated samples

S3DIS	2726	910	110	1166	274	-	-
Synthetic	364	252	86	-	-	200	192
Total	**3090**	**1162**	**196**	**1166**	**274**	**200**	**192**

3.2. Model training

This section provides an overview of the experiments conducted to investigate the influence of generated data on the model's performance. The details of these experiments are summarized in Table 2, which outlines the specific training approaches employed in each experiment, including the data utilized for both training and evaluation. As previously mentioned in the preceding section, the dataset is divided into nine distinct areas, with the final three being synthetically generated. All experiments were executed over five training epochs, and only the results from the best-performing epoch are presented in the Results section.

Table 2. Training experiments

Exp. ID	Description	Training Areas	Test Areas
1	Baseline. Original dataset, original 5 classes.	1, 2, 3, 4, 6	5
2	Mixed dataset, 5 classes (2 synthetic)	1, 2, 3, 7, 8	9
3	Mixed dataset, 5 classes (2 synthetic)	1, 2, 3, 4 ,6, 7, 8, 9	5

Experiment 1 serves as the baseline for assessing the influence of the generated data. In contrast, Experiment 2 aims to demonstrate the model's capacity to learn features resembling the new classes, making it the only experiment involving a distinct test area. Experiment 3 utilizes the entire set of generated data for testing against the fifth area, allowing for a comparison of performance metrics when classes are swapped.
The performance metrics employed in the experiments align with those specified in the original TR3D paper (Rukhovich *et al.*, 2023). The authors utilize mean average precision (mAP) calculated under Intersection over Union (IoU) thresholds of 0.25 and 0.5.
The implementation also follows the TR3D implementation details, utilizing the mmdetection3d framework (MMDetection3D, 2020). Both training and testing are conducted on a single Nvidia A4000 GPU. The training settings, including losses, optimizer, learning rate, and augmentation, adhere to the TR3D specifications, with the only deviation being the adjustment of batch size to accommodate the GPU CUDA memory constraints.
This quantitative model evaluation is complemented with a qualitative analysis in several real point cloud scans in the Results and discussion section.

3.3. Digital representation and CAD configurator

The CAD configurator employs two distinct information flow branches, both originating from the point cloud obtained via a LiDAR scanner and converging towards room generation (refer to Figure 1). In the first branch, the point cloud is input into the TR3D detector to identify the position of objects within the 3D space. Once these objects are aligned with the global coordinate system, their respective classes are linked to a CAD collection library and subsequently imported into the room assembly file.

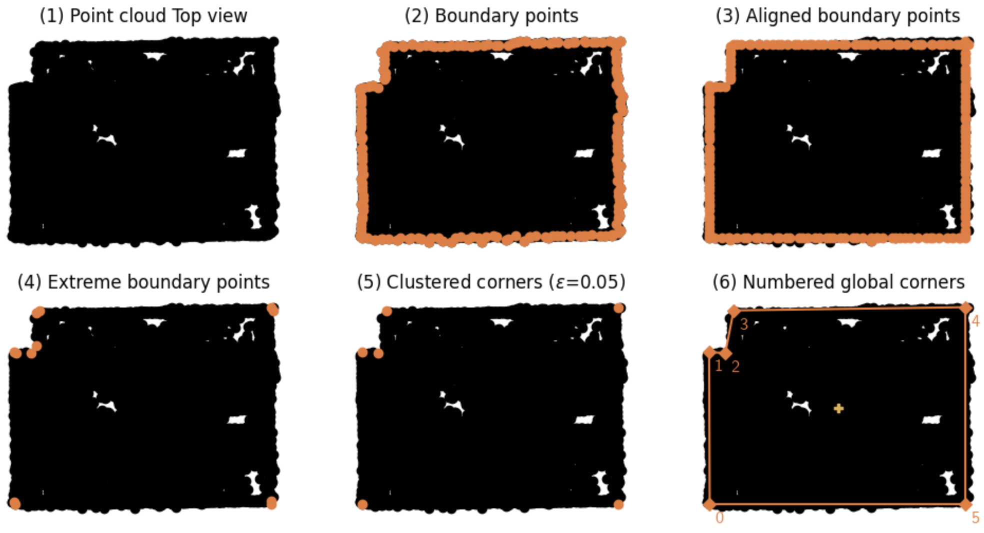
(1) Point cloud Top view
(2) Boundary points
(3) Aligned boundary points
(4) Extreme boundary points
(5) Clustered corners (ε=0.05)
(6) Numbered global corners
3
4
1
2
0
5

Figure 3. Corner detection steps; In (5), the parameter ε represents the normalized threshold for clustering points using DBSCAN; In (6), the corners are ordered in a clockwise manner according to the angle formed from the center of rotation

The second branch directly processes the point cloud data to delineate the scene's boundaries, resulting in an ordered array of points that can be utilized to sketch the floor within CAD software. While obtaining the scene's borders may appear to be a straightforward task, the Convex Hull method doesn't always adapt well to the room's geometry. As a result, a more intricate algorithm has been developed to accurately capture the blueprint's geometry. Figure 3 provides a step-by-step guide to identifying the room's corners sent to the configurator. It's worth noting that manual revision of the points may be implemented upon completion to correct any potential algorithmic discrepancies.
The completed configurator is made available to the user, enabling them to adjust the positions of CAD objects or manipulate geometric parameters within the CAD model. This digital representation empowers the user to delve deeper, explore, and refine the current 3D layout through simulations.

4. Results and discussion

This chapter unveils the experimental findings after training and testing. These results are substantiated by quantitative metrics that gauge the impact of integrating the proposed synthetic data generation in S3DIS. The qualitative outcomes are based on real data instances, serving as examples of the model's performance in real 3D scenes. These are not only compared against the baseline TR3D model but also illustrate the model's efficacy in detecting new classes that were solely processed as synthetic data during training. The concluding paragraphs are dedicated to discussing the framework as a digital representation configurator, highlighting observed strengths and limitations in the approach, and outlining potential future directions of work.

Table 3. Quantitative results; Testing results on S3DIS area 5 (except experiment 2 tested on area 9)

ID	Metric	🪑	⊤	🛋️	📚	📊	🦾	🖨️
1	mAP@0.25	0.96	0.72	0.87	0.37	0.11	-	-
	mAP@0.5	0.87	0.45	0.3	0.15	0.002	-	-
3	mAP@0.25	0.97	0.64	0.82	-	-	-	-
	mAP@0.5	0.89	0.42	0.46	-	-	-	-
2	mAP@0.25	1	1	1	-	-	1	1
	mAP@0.5	1	1	1	-	-	1	0.99

Table 3 presents quantitative results, specifically showcasing the mAP across different classes. The primary objective of these experiments is to evaluate the influence of synthetic data on the original S3DIS dataset. Experiment 1 serves as the baseline and demonstrates notably high accuracy, particularly in the case of chairs and tables. This heightened accuracy can likely be attributed to the substantial presence of elements belonging to these two classes within the dataset.
Experiment 2 aims solely to demonstrate the model's capacity to accurately identify the YuMI robot and 3D printers in the generated data. Impressively, it achieves perfect results for all classes in this regard. These results may be related to the fact that the dataset contains minimal noise, and the objects are exceptionally well-defined, as if the scanner used was a flawless tool.
Experiment 3 reveals how the model's testing results respond to the inclusion of synthetic data and the replacement of the bookcase and board classes with new ones. When analyzing this data, it's crucial to consider the quantity of synthetic data per class. Notably, the mAP@0.5 scores for chairs and tables exhibit only minor changes, as the added synthetic data accounts for just 11% and 21% of their respective datasets. Therefore, variations in the results can be attributed to training aspects.
In contrast, the mAP@0.5 for the sofa class demonstrates a noteworthy 53% improvement compared to the baseline model, which is a substantial increase. This improvement can be linked to the substantial representation of synthetic sofas (44%) in the dataset.

Table 4. Qualitative results; Ground truth for the number of items and those correctly (and incorrectly) recognized by tr3d trained on S3DIS (Baseline) and incorporating synthetic data (TR3D-sd); Threshold = 0.3

Scan ID - Quality	**Model**							
1 - Good	Truth	6	1	0	0	0	0	0
	Baseline	6(0)	1(0)	0(0)	0(3)	0(0)	-	-
	TR3D-sd	6(0)	1(0)	0(0)	-	-	0(0)	0(0)
2 - Poor	Truth	1	3	0	0	1	1	0
	Baseline	1(0)	1(0)	0(0)	0(4)	0(0)	-	-
	TR3D-sd	1(0)	2(0)	0(1)	-	-	1(0)	0(0)
3 - Medium	Truth	0	2	0	1	0	1	1
	Baseline	0(0)	2(0)	0(0)	1(0)	0(0)	-	-
	TR3D-sd	0(0)	2(0)	0(0)	-	-	1(0)	0(0)
4 - Medium	Truth	8	1	0	0	0	0	0
	Baseline	8(0)	1(0)	0(0)	0(3)	0(0)	-	-
	TR3D-sd	8(0)	1(0)	0(0)	-	-	0(0)	0(0)
5 - Poor	Truth	3	6	0	0	0	0	3
	Baseline	3(2)	3(0)	0(0)	0(0)	0(0)	-	-
	TR3D-sd	3(3)	4(0)	0(0)	-	-	0(0)	0(2)
6 - Good	Truth	33	8	0	0	2	0	0
	Baseline	23(0)	8(0)	0(0)	0(3)	0(0)	-	-
	TR3D-sd	33(0)	8(0)	0(0)	-	-	0(0)	0(0)

At a broader level, three key conclusions can be drawn from the qualitative results:

1. The quality of the scans significantly influences all objects detection. Without the support of colors, certain objects may go unnoticed to the human eye, such as the 3D printers in room 5.
2. Both algorithms exhibit strong performance in detecting tables and chairs, underscoring the model's potential when trained on a dataset with over 1000 samples. Enhancing the representation of other objects through synthetic data has the potential to further improve the model's performance.
3. The model augmented with synthetic data demonstrates remarkable accuracy in identifying the YuMI robot, likely due to its distinctive shape. Conversely, the model struggles to identify 3D printers in real point clouds, while achieving perfect accuracy in the generated dataset. This discrepancy suggests a clear case of overfitting, which may necessitate improvements in the generator or training on real samples for resolution.

Regarding the room configurator, it has the capability to identify detected objects and place them within an automatically generated room. It's important to note that the concept of room recreation is not new, and commercial solutions like Polycam already offer integrated solutions for this purpose. However, our proposal involves the development of such a system within a CAD configurator, which offers dynamic, customizable, and parametric geometries. This system can be further manipulated to replicate real factory layouts, thus significantly reducing the need for manual reconstruction.

Future directions for research primarily revolve around the development of the automatic synthetic data generator. Enhancing this generator could play a pivotal role in advancing the training and overall progress of 3D computer vision. Potential avenues for improving synthetic data generation may encompass deliberate noise injection, the integration of additional artificial samples and objects to simulate "clutter," adjustments to the objects-to-background point proportion, inclusion of more object models, and establishing connections with projects like ShapeNet or ModelNet40 to enhance generalization. Other possibilities include the release of an API and expansion to encompass outdoor environments and scenery incorporating Helios++ (Winiwarter *et al.*, 2021) simulation framework.

5. Conclusion

The presented research emphasizes the significance of employing automatic synthetic data generation in data-driven applications. Specifically, the paper focuses on the development and assessment of a 3D computer vision system for constructing digital representations of 3D environments within engineering contexts, such as brownfield factory projects. Key facets of this work include:

- The utilization of synthetic data to enhance the performance of 3D object detection and recognition.
- The successful development of an automatic synthetic data generator and its integration with the S3DIS dataset, showcasing notable improvements in object detection.
- Experiments to assess the impact of synthetic data, including the identification of specific object classes.
- The significance of high-quality scans and their influence on object detection.
- Potential areas for improvement and future research, such as refining the synthetic data generator, introducing noise, enhancing generalization, and exploring outdoor scenarios.
- The proposal of a room configurator within a CAD environment for efficient and customizable room recreation.

In conclusion, this work demonstrates the potential of synthetic data in enhancing 3D computer vision applications in an engineering context. It highlights the importance of data quality and presents promising avenues for further research and development, particularly in the realm of synthetic data generation and its broader applications.

Acknowledgements

This work was supported by Vinnova under grant 2021-02481, iPROD project; and grant 2023-02694, Twin Generator project. The authors would like to thank Vinnova for making this research project possible.

References

ABB. (2015), “Dual-arm YuMi - IRB 14000”, ABB.

Anderson, J.W., Kennedy, K.E., Ngo, L.B., Luckow, A. and Apon, A.W. (2014), “Synthetic data generation for the internet of things”, *2014 IEEE Big Data*, pp. 171–176, https://dx.doi.org/10.1109/BigData.2014.7004228.

Armeni, I., Sax, A., Zamir, A.R. and Savarese, S. (2017), “Joint 2D-3D-Semantic Data for Indoor Scene Understanding”, *2017 IEEE CVPR*, https://dx.doi.org/10.48550/arXiv.1702.01105.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., *et al.* (2015), “ShapeNet: An Information-Rich 3D Model Repository”, arXiv, https://dx.doi.org/10.48550/arXiv.1512.03012.

Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M. and Yin, B. (2018), “Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges”, *IEEE Access*, Vol. 6, pp. 6505–6519, https://dx.doi.org/10.1109/ACCESS.2017.2783682.

Chen, W., Li, Y., Tian, Z. and Zhang, F. (2023), “2D and 3D object detection algorithms from images: A Survey”, *Array*, Vol. 19, p. 100305, https://doi.org/10.1016/j.array.2023.100305.

MMDetection3D, C. (2020), “MMDetection3D: OpenMMLab next-generation platform for general 3D object detection”.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T. and Nießner, M. (2017), “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”, *2017 IEEE CVPR*, https://dx.doi.org/10.48550/arXiv.1702.04405.

Esteves, C., Allen-Blanchette, C., Makadia, A. and Daniilidis, K. (2017), “Learning SO(3) Equivariant Representations with Spherical CNNs”, *2017 IEEE CVPR*, https://dx.doi.org/10.48550/ARXIV.1711.06721.

Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T. and Wong, E. (2015), “3D deep shape descriptor”, *2015 IEEE CVPR*, pp. 2319–2328, https://dx.doi.org/10.1109/CVPR.2015.7298845.

Blender, F. (n.d.). “Raycast Node”, Blender 3.6 Manual.

Martínez, G.S., Karhela, T.A., Ruusu, R.J., Sierla, S.A. and Vyatkin, V. (2018), “An Integrated Implementation Methodology of a Lifecycle-Wide Tracking Simulation Architecture”, *IEEE Access*, Vol. 6, pp. 15391–15407, https://dx.doi.org/10.1109/ACCESS.2018.2811845.

Maturana, D. and Scherer, S. (2015), “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”, *2015 IEEE/RSJ IROS*, pp. 922–928, https://dx.doi.org/10.1109/IROS.2015.7353481.

Nguyen, H.G., Habiboglu, R. and Franke, J. (2022), "Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing", *Procedia CIRP*, Vol. 107, pp. 1263–1268, https://doi.org/10.1016/j.procir.2022.05.142.

Piascik R., et al. (2010), "Technology Area 12: Materials, Structures, Mechanical Systems, and Manufacturing Road Map", NASA Office of Chief Technologist.

Qi, C.R., Litany, O., He, K. and Guibas, L.J. (2019), "Deep Hough Voting for 3D Object Detection in Point Clouds", *2019 ICCV*, https://dx.doi.org/10.48550/arXiv.1904.09664.

Qi, C.R., Su, H., Mo, K. and Guibas, L.J. (2017), "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", *2017 IEEE CVPR*, https://dx.doi.org/10.48550/ARXIV.2302.02858.

Qi, C.R., Yi, L., Su, H. and Guibas, L.J. (2017), "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space", *ArXiv CVPR*, arXiv, https://dx.doi.org/10.48550/ARXIV.2302.02858.

Regenwetter, L., Curry, B. and Ahmed, F. (2021), "BIKED: A Dataset for Computational Bicycle Design With Machine Learning Benchmarks", *Journal of Mechanical Design*, Vol. 144 No. 3, p. 31706, https://dx.doi.org/10.1115/1.4052585.

Rukhovich, D., Vorontsova, A. and Konushin, A. (2023), "TR3D: Towards Real-Time Indoor 3D Object Detection", *ArXiv CVPR*, https://dx.doi.org/10.48550/ARXIV.2302.02858.

Schluse, M., Priggemeyer, M., Atorf, L. and Rossmann, J. (2018), "Experimentable Digital Twins—Streamlining Simulation-Based Systems Engineering for Industry 4.0", *IEEE TII*, Vol. 14 No. 4, pp. 1722–1731, https://dx.doi.org/10.1109/TII.2018.2804917.

Shellshear, E., Berlin, R. and Carlson, J.S. (2015), "Maximizing Smart Factory Systems by Incrementally Updating Point Clouds", *IEEE CGA*, Vol. 35 No. 2, pp. 62–69, https://dx.doi.org/10.1109/MCG.2015.38.

Sierla, S., Azangoo, M., Fay, A., Vyatkin, V. and Papakonstantinou, N. (2020), "Integrating 2D and 3D Digital Plant Information Towards Automatic Generation of Digital Twins", *2020 IEEE ISIE*, pp. 460–467, https://dx.doi.org/10.1109/ISIE45063.2020.9152371.

Sierla, S., Sorsamäki, L., Azangoo, M., Villberg, A., Hytönen, E. and Vyatkin, V. (2020), "Towards Semi-Automatic Generation of a Steady State Digital Twin of a Brownfield Process Plant", *Applied Sciences*, Vol. 10 No. 19, https://dx.doi.org/10.3390/app10196959.

Song, S., Lichtenberg, S.P. and Xiao, J. (2015), "SUN RGB-D: A RGB-D scene understanding benchmark suite", *2015 IEEE CVPR*, pp. 567–576, https://dx.doi.org/10.1109/CVPR.2015.7298655.

Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E. (2015), "Multi-view Convolutional Neural Networks for 3D Shape Recognition", *ArXiv CVPR*, arXiv, https://dx.doi.org/10.48550/ARXIV.1505.00880.

Wang, X., Pan, H., Guo, K., Yang, X. and Luo, S. (2020), "The evolution of LiDAR and its application in high precision measurement", *IOP EES*, IOP Publishing, Vol. 502 No. 1, p. 12008, https://dx.doi.org/10.1088/1755-1315/502/1/012008.

Winiwarter, L., Pena, A.M.E., Weiser, H., Anders, K., Sanchez, J.M., Searle, M. and Höfle, B. (2021), "Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic 3D laser scanning", *arXiv CVPR*, https://dx.doi.org/10.48550/arXiv.2101.09154.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. (2015), "3D ShapeNets: A deep representation for volumetric shapes", *2015 IEEE CVPR*, pp. 1912–1920, https://dx.doi.org/10.1109/CVPR.2015.7298801.