2022 European Conference on Computing in Construction
Ixia, Rhodes, Greece
July 24-26, 2022

GENERATING GEOMETRIC DIGITAL TWINS OF BUILDINGS: A REVIEW

Viktor Drobnyi, Yasmin Fathy, and Ioannis Brilakis
University of Cambridge, Cambridge, UK

ABSTRACT

Generation of geometric Digital Twins of existing buildings relies on point cloud datasets and is still a manual-intensive and time-consuming process. This paper identifies the most frequent object types in buildings, analyses how current commercial software and state-of-the-art research methods to generate these objects. We summarise the main advantages of these methods and highlight limitations that limit these methods from broader adoption by the industry. Later, we identify the open challenges and discuss future directions to enable automating geometric Digital Twin generation.

INTRODUCTION

Digital Twins (DTs) are digital replicas of real-world buildings, which integrate real-time data from physical objects to a virtual environment to improve the building operation (Tao et al., 2019). Geometric Digital Twin (gDT) is a geometry representation of a DT. Generating gDT refers to generating individual objects of a Physical Twin and relations between these objects.

More than half of buildings currently in service in the UK were built more than 20 years ago (GOV.UK, 2020, 2021). Their design models are not available (Enterprises, 2020), and often their physical assets are not fully representative (Mahdjoubi et al., 2015); therefore, PCD (Point Cloud Dataset) has been used for creating DTs. A growing number of buildings and infrastructure are currently being repaired, while others will soon require major renovations. It is estimated that 85-95% of buildings will remain in use by 2050 (Commission", 2021), many of them will be renovated. Some of the older buildings were constructed prior to the age of emerging digital technologies where their as-designed models are unavailable; they do not have as-built and as-designed digital models.

The generation of DTs for existing buildings when as-designed or as-built models are unavailable is a manual, labour-intensive process whose labour costs are perceived to counteract the value of the resulting DT (Hossain and Yeoh, 2018; McArthur, 2015). In principle, every square metre of a building takes up to 5 minutes of labour time (Qu et al., 2014), leading to months of work for generating geometry from a PCD. For example, manual geometry generation for 420 square metres of a pumping facility took 166 hours (Qu et al., 2014). Commercial software tools for DTs generation such as Revit do not provide automatic geometry generation from PCDs. This indicates the lack of automation in geometric DT generation. According to the NBS survey on BIM adoption by the industry, 46% of British professionals in the construction sector reported that high costs and time consumption were barriers to creating data enriched building models for newly designed assets (Enterprises, 2020). The portion of those who adopt it for already built facilities is notably smaller based on this study.

The value of automating geometry generation stems from the fact that the geometry of a DT is a core part of the building data and a prerequisite in the broader asset digitisation process. DTs are the outcome of digitisation and have the potential to improve the efficiency of many processes in building management, maintenance and renovation. Automating the generation of DT geometry (geometric Digital Twin or gDT) of already built buildings from PCD is essential to facilitate the further digitisation of the construction sector, especially for old buildings to plan for deep renovation and/or major repairs.

PROBLEM DEFINITION

It is essential to identify the most frequent components of buildings to understand what object types need to be generated to digitise building's geometry. The authors analysed sufficiently complete building models on the design, construction and operation phases to identify object types that appear more often than the others. The purpose of the frequency analysis was to narrow down further analysis to object types that are more likely to appear during geometry generation but not to identify average frequencies with high precision.

The dataset of 24 Industry Foundation Classes (IFC) models consisted of 105k total object counts were gathered for the analysis. These models represented a variety of buildings, including office buildings, university's buildings, hospitals and residential buildings. Table 1 shows ten most frequent object types in buildings in the IFC dataset. Together, these object types account for almost 80% of all objects from the dataset. This entails that the majority of the effort for the geometry digitisation of a building will be spent on the generation of one of these ten object types.

BACKGROUND

The generation of the geometry of a DT when an as-designed model is unavailable relies on PCDs, which

Table 1: Top 10 most frequent object types in buildings.

Object type	Fraction(%)
Round Pipe Segment	23.43
Wall	10.87
Round Elbow	10.85
Beam	7.82
Column	5.02
Slab	5.02
Light Fixture	4.52
Plumbing or Heating Terminal	4.04
Cylindrical Joint	3.88
Rectangular Duct Segment	3.44

are a set of points. Each point is represented by its Cartesian coordinates and their colours (or intensities) sampled from a real-world object. This representation of geometry is not directly usable for DTs because it lacks semantics and structure. The application of DTs requires a highly structured representation of geometry. This includes decomposing the space into individual objects with their geometry representation and relations between them. If the as-designed models are not available, which is the case for most old buildings, we can not infer the decomposition of a PCD from the design intent. This entails that a pre-processed point cloud should be enriched by the semantic and object relations such that the pre-processed PCD should be decomposed into objects with their geometry and relations between these objects.

The geometrical properties of every type of object can be described by its shape, colour and the context in which it appears. The geometry of most of the frequent object types in buildings, such as walls, slabs and pipes, is a combination of planar or cylindrical patches. However, the geometry of some other objects, such as beams, can also be described as an extrusion of a profile along a line. For example, I-beam can be described as a combination of 8 (or 12) planar patches or an I-shaped 2D profile swept along its axis. The geometry of a building can also be viewed as a hierarchy of spaces and objects. For example, a building usually consists of multiple stories. Stories have rooms and corridors; rooms have wall surfaces, doors, furniture; doors have knobs, etc.

The following section describes current commercial software for generating gDTs from PCDs, its current functionality and limitations. The later section briefly discusses the state of research in this area. Finally, the discussion section summarises the capabilities of the state of research and states the main challenges for future research for generating gDTs from PCDs automatically.

STATE-OF-THE-ART IN PRACTICE

The state-of-the-art software for gDT generation from PCDs aim to assist a human modeller by providing convenient tools to detect shapes in PCDs, isolate PCDs of individual objects and fit objects into them. Examples of these software include "EdgeWise", "Faro as-built modeller", "Pointfuse", "Scantobim.xyz", "InfiPoints", "Point Cab", "Vision Lidar", "Leica Cyclone Model" and "Leica Cloudworx", "E3D Design", "Trimble Realworks", among others. Many of these software were initially designed for industrial assets and, therefore, have richer functionality to generate piping networks since piping elements are more important for plants (Agapaki et al., 2018).

Some of these software provides the functionality to detect planar patches automatically. These planar surfaces can then be used to model walls, slabs, etc. For example, "Pointfuse" can detect and classify planes into wall surfaces, slab surfaces and other objects. The differentiation between classes is based on normal orientation; it considers that vertical planes are parts of walls and horizontal planes are parts of floors and ceilings (Figure 1). Other software, such as "EdgeWise", can automatically combine detected vertical planes into parallelepipeds to generate walls (Figure 2).

Figure 1: Automatic wall surface extraction and classification in PointFuse. A human modeller needs to manually combine these surfaces and manually introduce new surfaces to generate wall objects.

Figure 2: Automatic wall extraction in "EdgeWise". The set of detected walls is incomplete and contains wrong cuboids. Besides, the accuracy of wall dimensions is not perfect and a human needs to adjust wall length manually.

Similarly, "EdgeWise" detects cylinders in a PCD to detect piping networks. It automatically detects cylinders in point clouds that are then used as pipe segments. Then, pipe endings that are closely located are detected, as they are most likely to be connected; based on that, pipe fittings are proposed. The quantitative study shows that EdgeWise achieves 62% precision and 75% recall in detecting cylinders (Agapaki et al., 2018).

Other software, such as "Faro as-built", use the iterative semi-automatic approach to detect pipe segments and fittings. This requires a user to pick a point on one of the pipes in the PCD. The software then automatically detects a cylinder that this point belongs to. After that, the software searches for the next piping element on the end of the detected pipe. This process iterates until no new pipes or fittings can be detected. Manual intervention during each iteration is necessary to adjust the fitting and its classification of elements from a catalogue.

Commercial software also use human involvement to detect and generate shapes of extrusion. In the first step, a user should specify a plane containing the profile, which should be perpendicular to the extrusion axis. After that, the software automatically detects and classifies the profile (Figure 3) and finds the appropriate length of the object. Lastly, if objects of the same type and shape are located in a regular grid, some software may use this information to automatically detect copies of a modelled object (such as "EdgeWise").

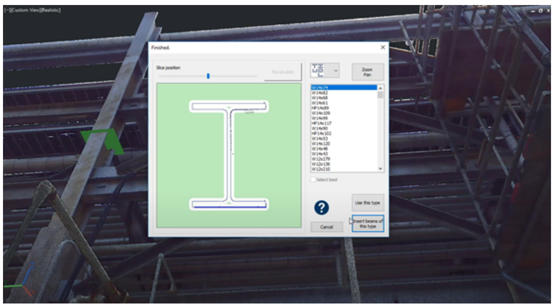

Figure 3: I-beam profile fitting in "Faro as-built"

Template matching is another approach to detect objects that have a less regular structure. This usually requires a user to select a PCD that represents one of the objects manually. Then the software can automatically detect similar PCD patches. The main shortcoming of this approach is that it works for relatively small PCDs or when the search space is limited. For example, "EdgeWise" allows selecting templates for door and window detection but requires to specify a plane on which they are located.

Objects with complex shapes, such as machinery, are often hard to detect. The generation requires coarse positions for fitting a given model into the PCD automatically. The input location can be provided in two ways: directly specifying a 3D point in a software's viewer or selecting a PCD part representing the given object.

Overall, commercial software for generating a building's geometry from a PCD can significantly reduce manual effort. The software are able to detect basic shapes and combine them into objects with relatively simple shapes, such as walls. However, there is still a substantial amount of manual work required. The user must verify every object detected automatically, manually adjust details (such as boundaries of objects), and generate objects that were not detected, which is time-consuming.

STATE-OF-THE-ART IN RESEARCH

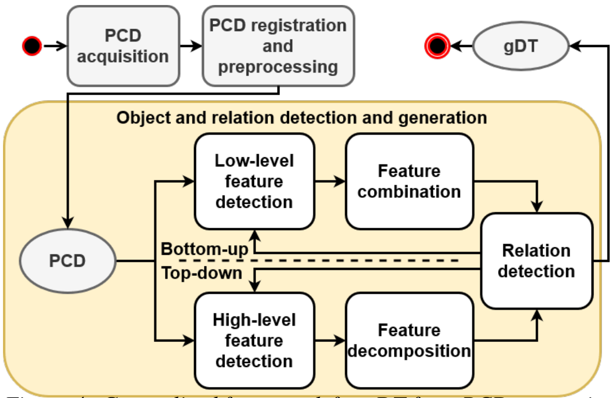

Figure 4: Generalised framework for gDT from PCD generation

Researchers have attempted to step beyond the state of practice and automate the generation of main object types for buildings. While some attempt to achieve superb quality in detecting and generating objects, others try to formulate the problem in broader terms and reconstruct buildings in terms of more high-level abstractions, such as available spaces and floor plans.

Object detection approaches in PCDs can be classified into two groups: bottom-up and top-down (Figure 4). Methods of the first group start from detecting local features, such as planar patches, cylinders, corners, etc. After that, primitives are combined to form building elements. This group of methods represents bottom-up approaches because they create a DT going from local features and combine elements to form the final DT. In contrast, methods that start from detecting high-level features and iteratively split a PCD into a set of lower-level features, such as going from floors to rooms to walls to doors, are top-down approaches because they gradually decompose a building into smaller and smaller elements. The following sections will discuss each group.

Bottom-up

The bottom-up approaches require finding primitives and detecting local features. One of the most popular methods that find primitives in PCDs is RANSAC (Fischler and Bolles, 1981) and its variants, for example, multiBaySAC (Kang and Li, 2015). This algorithm randomly samples a minimum number of points to form a hypothesis, verify the hypothesis (i.e. how many points from the input align with this hypothesis - inliers), and repeat the process to maximise the number of inliers. RANSAC can be used to detect planar patches or cylinders in PCDs to detect objects. For example, this has been successfully applied to detect wall surfaces (Anagnostopoulos, Patraucean, Brilakis and Vela, 2016). Later, the authors detected connections between perpendicular walls to adjust wall lengths by enlarging them to touch each other and form rooms (Anagnostopoulos, Belsky and Brilakis, 2016). Interested readers in RANSAC-based methods can be referred to the review of Raguram et al. (2008).

The same goal can be achieved using Hough transform for primitive shape detection (Hough, 1959). This algorithm goes through each point from the input and computes all possible hypotheses that explain this point, and picks the most frequent hypothesis. It is also used to detect primitive geometry. For example, this can be used to detect cylinders for pipe generation (Patil et al., 2017). Both of these methods require a tolerance threshold to be defined; they are sensitive to clutter and occlusions and are hardly scalable due to computational inefficiency on large inputs.

Planar and cylindrical surfaces have homogeneous or constant local curvature. For instance, a plane has zero curvature, while pipes' curvature equals the pipe's radius. A PCD with curvature can be clustered to detect various homogeneous patches. For example, a region growing algorithm can be used to detect planar and cylindrical surfaces Dimitrov and Golparvar-Fard (2015). This algorithm can handle larger PCDs but is more sensitive to noise.

Many objects have more complicated shapes, such as I-beams or machinery, and the methods mentioned above cannot be directly used to detect them. Wang et al. (2021) proposed the method to search for objects that have a geometry of a 2D profile extruded along a line. They suggested slicing the PCD along an axis and searching for the desired profile in each slice (Figure 5). This converts the problem from 3D to 2D; similar methods are used to detect lines and circles on a plane. The authors showed that this approach could find various objects of extrusion with more complex shapes, such as I-beams, rectangular ducts, etc. In addition, they showed that template matching in 2D is a computationally feasible problem, and it can be used to match arbitrary profiles to detect arbitrarily shaped machinery. The method requires objects to be oriented parallel to main axes only.

The alternative approach to detect objects with complex shapes is to combine primitives together to form object surfaces. This includes using predefined rules to connect surfaces together, such as combining nearly parallel vertical planes located near each other to form a wall. Another approach is to use data-driven methods to

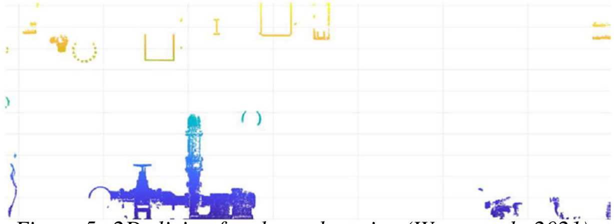

Figure 5: 2D slicing for shape detection (Wang et al., 2021). The algorithm searches for a profile shape in the slice to detect a shape of extrusion or template to detect complex machinery.

construct parametric models from the set of primitives, for example, using random forests (Zhang et al., 2014).

Recent advancements in deep learning allow using neural networks to perform PCD understanding. Using supervised learning, neural networks are used to split PCDs with multiple objects of different types inside into point clusters of particular object types or individual objects. Later, these point clusters can be transformed into meshes or predefined models from a catalogue using RANSAC or similar methods. A few examples would be (Chen et al., 2019; Armeni et al., 2016; Thomas et al., 2019).

Supervised machine learning methods infer dependencies and learn features from labelled data. The main benefit is it avoids exploiting any explicit knowledge of design and construction practices and patterns. However, this makes neural networks require large labelled datasets to be trained to achieve high accuracy of label prediction.

Top-down

Top-down approaches aim to detect high-level objects in the input PCD, starting from floors to rooms to walls to doors to windows.. In the first step, they identify floors and split PCD into multiple smaller ones. The straightforward approach for ceiling and floor identification is to identify peaks along the Z-axis. This assumes that the direction of the gravity is known and Z-axis is collinear with it. Then, all points are projected on Z-axis and split into buckets to form a histogram of a number of points on multiple heights. Given that points sufficiently cover floors and ceilings, they should be located on the peaks of the histogram. The height of peaks and the knowledge of the approximate width of slabs and floor height allow identifying each floor and ceiling.

In the next step, rooms can be identified independently in each PCD, representing an individual floor. There are multiple approaches on how to locate rooms. Macher et al. (2017) proposed identifying rooms on the storey by computing a discrete occupancy map for the horizontal slice containing the ceiling followed by a region growing algorithm (Figure 6). This approach assumes that all spaces on a storey are disconnected on the top. This assumption holds if doors and other transitions between

Table 2: Summary of gDT generation approaches

Detector	Approach	Examples	Advantages	Limitations
Shape detector	Primitive shape detection	(Anagnostopoulos, Patraucean, Brilakis and Vela, 2016), (Patil et al., 2017)	Theoretically extensible on arbitrary shapes; robust to clutter and occlusions	Inefficient on large inputs or when large number of objects present
	Region-growing	(Dimitrov and Golparvar-Fard, 2015)	Robust, scalable	Over-segmentation; limited number of shapes
	2D Slicing and projection	(Wang et al., 2021)	Objects of extrusion, arbitrary and complex shapes	Objects should be located along the limited number of axes
	Deep learning supervised PCD segmentation	(Chen et al., 2019), (Thomas et al., 2019)	Need *only* labeled data to generalise	Need large set of labeled *data* to generalise
Space detector	Histograms of #points	(Huber et al., 2011), (Tran et al., 2018)	Simple	Intolerant to clutter and occlusions
	Floor-plan reconstruction	(Macher et al., 2017), (Ochmann et al., 2019)	Structured, connected output	Rely explicitly on knowledge, design patterns; hardly extensible

spaces are lower than the ceiling level. Shi et al. (2019) suggested computing alpha-shapes of rooms of the ceiling slice to adjust room shapes. This decomposition allows splitting the input into multiple PCDs, each representing only one room. Then these PCDs can be processed independently for the sake of computational efficiency.

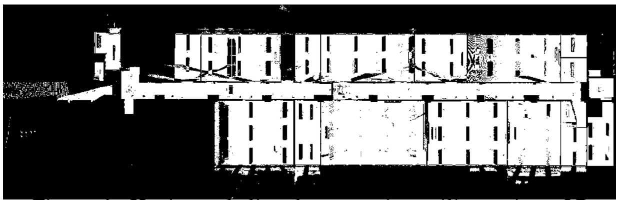

Figure 6: Horizontal slice that contains ceiling points. 2D room decomposition (Macher et al., 2017)

Another approach to computing room spaces on 2D projections is to project only boundary points. This is achieved by keeping only those points that have a normal perpendicular to Z-axis. The majority of points satisfying these conditions represent wall surfaces. However, this method is sensitive to wall occlusions and clutter with vertical surfaces.

Ochmann et al. (2016) proposed to segment a PCD into rooms based on the scan locations. They assumed that one scan of a laser scanner always captures one room. Scans that captured the same room are merged based on co-visibility obtained with ray casting (Ochmann et al., 2019), points are classified with room IDs based on the scan they belong to. Then, they computed occupancy bitmaps and produced floor plans. After that, they detected vertical planes, projected them to the floor plan and generated wall candidates based on a pair of close parallel planes. At the last step, they formulated the wall location problem as an optimisation problem: they penalised a wall candidate if it split one room and rewarded if it split different rooms. This method requires scanner locations and does not work for PCDs captured with SfM (as scanner location changes continuously).

A similar objective can be achieved using the shape grammar approach (Tran et al., 2018). The space is split into cubic volumes based on histogram peaks along the main axes. They then iteratively merged cuboids representing the same space and classified cuboids representing walls. However, the last step required the user to specify door locations to introduce space connections manually.

Void-growing approach for space detection was proposed by (Pan et al., 2021). The authors tried to find empty regions of a PCD by growing a cuboid until it touches a wall or slab. At the first stage, the authors searched for vertical planes to produce room centre candidates (they split space by planes parallel to two main axes and got cube centres). Then, they enlarged each cuboid in each direction until the points on the boundary of the cuboid face occupied a significant part of the face. In the final step of space detection, they discarded thin cuboids (walls) and merged overlapping spaces to account for non-rectangular rooms, such as L-shaped rooms.

The next step is decomposing PCDs that represent individual rooms into various objects, such as walls and openings in walls. Shi et al. (2019) proposed automatically detecting doors by searching for empty spaces on wall surfaces. They computed an occupancy map on each wall surface to get empty spots on the surface. Then they identified those voids that might present due to occlusions and identified doors. This method is limited to open doors only. Quintana et al. (2018) based their door search on empty regions and rectangular objects on wall texture. They assumed that a door has a different colour than the wall on which it is located; therefore, they warped wall points onto the plane and searched for rectangles using a Canny edge detector. The drawback of this method is its sensitivity to wall detection, wall occlusions, lighting conditions and the colour difference between walls and doors.

DISCUSSION

The state-of-practice software simplifies the generation of the most frequent object types in buildings to some extent. The software can automatically detect primitive shapes, such as planar patches and cylinders in PCDs and provide a user interface to generate objects and connect them to create DTs. These tools significantly reduce the manual effort necessary for digitising the geometry of existing assets from PCDs.

However, the generation of each object still requires manual involvement. Objects with planar surfaces require manual adjustments of dimension sizes. For example, automatically generated walls and pipes in "EdgeWise" require a manual extension (or shortening) along the length. Automatic pipe fitting generation in this software considers only pipe endings and does not account for the PCD itself. It results in objects with wrong parameters (e.g. elbows with the wrong radius). Semi-automatic pipe run detection implemented in "Faro", which tries to automatically fit the next network element, requires manual involvement for each object. The same holds for the detection of steel structures.

The state-of-the-art software can detect primitive shapes automatically or fit objects from a catalogue (precisely defined) semi-automatically. It can also detect relations between parts of pipe runs. However, users still need to do manual work to adjust automatically detected objects, guide semi-automated detection and detect connections.

On the other hand, state of the art in research can detect primitive shapes automatically and combine them to generate objects. Bottom-up approaches provide similar functionality to the mentioned functionality of software. It gives more agility, allows to detect more primitive shapes, and combines them with explicit rules and machine learning, adjusting object dimensions and connecting them together. However, this group of methods is limited in generating composite structures due to high variability. On the other hand, top-down approaches naturally provide hierarchical relations between objects. These relations are usually required for DT applications.

Most of the existing methods for generating gDTs from PCDs are designed to detect particular types of objects or shapes. These methods are deterministic rule-based. They use many assumptions about object relations or design patterns to guide detection, such as the Manhattan-world assumption. The explicitly exploited knowledge in an algorithm limits the adoption of the algorithm to other shapes and contexts.

Summing up, the functionality of commercial software and state of research automatically detect some of the object types and compose (or decompose) them to some extent. To the best of our knowledge, there are no methods dedicated to detecting relations. Table 2 provides a brief summarisation of available methods for detecting different objects of buildings in PCDs. We split the existing methods based on the detection approach: shape detection and space detection. Methods for shape detection based on detecting local features, such as RANSAC and region growing algorithms, are limited in terms of the variability of shapes it can detect. The reason is that these methods can only detect geometrically homogeneous regions. Besides, they require extra steps to combine multiple surfaces together.

Methods projecting 3D PCDs onto smaller dimensions reduce the complexity of the detection problem but are suitable only for a subset of objects such as that objects of extrusion. They assume that the orientation of objects is known to some degree. This is because they project points on a limited number of axes or planes. Otherwise, the method would be computationally infeasible.

Methods that are based on space decomposition tend to produce much more connected and structured output, which is more useful for DT applications. On the other side, these methods typically encode some knowledge or assumptions about a building explicitly. It makes these methods good in detecting and generating particular structures but hardly extensible to buildings with other design choices. This is particularly true for systems with high variability: while these types of methods are used to generate the architectural part of a building, they are not used to generate the mechanical part.

Some deep learning approaches have been discussed previously in this paper. These methods extract the implicit knowledge about buildings through observations. This entails that the extension to other environments with other design patterns only requires labelled data representing target distribution. The main drawback is that it requires large labelled datasets in order to generalise well. Semi-supervised methods or generating synthetic datasets could be potentially considered to address this challenge.

Suggestions for the future research

The authors identify the following challenges that should be tackled to develop an effective method for automatic gDT generations:

- How to make the methods generic enough to be extended to other contexts and environments easily? Authors believe that successful automation of geometry digitisation requires more generic methods to detect and generate gDT. Current deterministic algorithms are hardly adaptable to environments that differ from the original assumption of the algorithms' authors. Supervised and semi-supervised deep learning are promising approaches to address these challenges.

- If adopting neural networks, how to sample PCDs from models to gather training data, how much synthetic data is necessary to generalise to unseen real-world data? Adopting neural networks is associated with crafting large labelled datasets for training, which is hardly feasible for real-world data. Alternatively, researchers can use synthetic datasets sampled from as-designed models to train neural networks.
- How to detect relations between objects? How to use them to empower object detection? Authors believe that current-of-the-art methods lack relation detection part. Many DT applications also require knowing what objects are related to each other and how they influence other objects. Besides, information about object relations may guide object detection. For example, search space for wall detection can be limited, knowing that walls related to room spaces as their boundaries.

CONCLUSIONS

Digitising the geometry for existing buildings heavily relies on PCDs due to the absence of reliable, designed models. The digitisation process is still semi-manual and requires substantial manual effort, which limits the adoption of DTs for the operational phase of the building life cycle. State of the art is limited in the variety of objects, relations and their contexts. The industry will benefit from the automation of gDT generation from PCDs.

This paper showed the most frequent object types in buildings that account for about 80% of all objects in buildings on average. We provided a review of the functionality of the available software, and current approaches for digitising geometry in research. It highlighted the advantages and limitations of different methods and proposed areas for future research. The authors believe that addressing the mentioned limitations will significantly reduce the cost and effort necessary for digitisation, renovation and maintenance of existing buildings.

ACKNOWLEDGEMENTS

The authors acknowledge the help of Zhiqi Hu in revising frequency analysis. This work is funded by the European Commission's Horizon 2020 for the CBIM (Cloud-based Building Information Modelling) European Training Network under agreement No. 860555 and partially supported by EU Horizon 2020 BIM2TWIN project under agreement No. 958398.

References

Agapaki, E., Miatt, G. and Brilakis, I. (2018), 'Prioritizing object types for modelling existing industrial facilities', *Automation in Construction* **96**, 211–223.

Anagnostopoulos, I., Belsky, M. and Brilakis, I. (2016), Object Boundaries and Room Detection in As-Is BIM Models from Point Cloud Data.

Anagnostopoulos, I., Patraucean, V., Brilakis, I. and Vela, P. (2016), 'Detection of Walls, Floors, and Ceilings in Point Cloud Data', pp. 2302–2311. Publisher: American Society of Civil Engineers.

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M. and Savarese, S. (2016), 3D Semantic Parsing of Large-Scale Indoor Spaces, pp. 1534–1543.

Chen, J., Kira, Z. and Cho, Y. K. (2019), 'Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction', *Journal of Computing in Civil Engineering* **33**(4), 04019027. Publisher: American Society of Civil Engineers.

Commission", E. (2021), 'Preliminary Analysis of the Long-Term Renovation Strategies of 13 Member States.'.

Dimitrov, A. and Golparvar-Fard, M. (2015), 'Segmentation of building point cloud models including detailed architectural/structural features and MEP systems', *Automation in Construction* **51**, 32–45.

Enterprises, N. (2020), 'NBS BIM Report 2020'.

Fischler, M. A. and Bolles, R. C. (1981), 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM* **24**(6), 381–395.

GOV.UK (2020), 'Non-domestic rating: stock of properties including business floorspace, 2020'.
URL: *https://www.gov.uk/government/statistics/non-domestic-rating-stock-of-properties-2020*

GOV.UK (2021), 'Construction statistics annual tables - Office for National Statistics'.
URL: *https://www.ons.gov.uk/businessindustryandtrade/constructionindustry/datasets/constructionstatisticsannualtables*

Hossain, M. A. and Yeoh, J. K. W. (2018), 'BIM for Existing Buildings: Potential Opportunities and Barriers', *IOP Conference Series: Materials Science and Engineering* **371**, 012051. Publisher: IOP Publishing.

Hough, P. V. C. (1959), 'Machine Analysis of Bubble Chamber Pictures', *Conf. Proc. C* **590914**, 554–558.

Huber, D., Akinci, B., Adan, A., Okorn, B. and Xiong, X. (2011), 'Methods for Automatically Modeling and Representing As-built Building Information Models'.

Kang, Z. and Li, Z. (2015), 'Primitive Fitting Based on the Efficient multiBaySAC Algorithm', *PLOS ONE* **10**(3), e0117341. Publisher: Public Library of Science.

Macher, H., Landes, T. and Grussenmeyer, P. (2017), 'From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings', *Applied Sciences* **7**(10), 1030. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Mahdjoubi, L., Brebbia, C. A. and Laing, R. (2015), *Building Information Modelling (BIM) in Design, Construction and Operations*, WIT Press. Google-Books-ID: uBJmCgAAQBAJ.

McArthur, J. J. (2015), 'A Building Information Management (BIM) Framework and Supporting Case Study for Existing Building Operations, Maintenance and Sustainability', *Procedia Engineering* **118**, 1104–1111.

Ochmann, S., Vock, R. and Klein, R. (2019), 'Automatic reconstruction of fully volumetric 3D building models from oriented point clouds', *ISPRS Journal of Photogrammetry and Remote Sensing* **151**, 251–262.

Ochmann, S., Vock, R., Wessel, R. and Klein, R. (2016), 'Automatic reconstruction of parametric building models from indoor point clouds', *Computers & Graphics* **54**, 94–103.

Pan, Y., Braun, A., Brilakis, I. and Borrmann, A. (2021), 'Void-growing: a novel scan-to-bim method for manhattan world buildings from point cloud'.

Patil, A. K., Holi, P., Lee, S. K. and Chai, Y. H. (2017), 'An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds', *Automation in Construction* **75**, 65–78.

Qu, T., Coco, J., Rönnäng, M. and Sun, W. (2014), 'Challenges and Trends of Implementation of 3D Point Cloud Technologies in Building Information Modeling (BIM): Case Studies', pp. 809–816. ISBN: 9780784413616 Publisher: American Society of Civil Engineers.

Quintana, B., Prieto, S. A., Adán, A. and Bosché, F. (2018), 'Door detection in 3D coloured point clouds of indoor environments', *Automation in Construction* **85**, 146–166.

Raguram, R., Frahm, J.-M. and Pollefeys, M. (2008), A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus, *in* D. Forsyth, P. Torr and A. Zisserman, eds, 'Computer Vision ECCV 2008', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 500–513.

Shi, W., Ahmed, W., Li, N., Fan, W., Xiang, H. and Wang, M. (2019), 'Semantic Geometric Modelling of Unstructured Indoor Point Cloud', *ISPRS International Journal of Geo-Information* **8**(1), 9. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Guo, Z., Lu, S. C.-Y. and Nee, A. Y. C. (2019), 'Digital twin-driven product design framework', *International Journal of Production Research* **57**(12), 3935–3953. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2018.1443229.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F. and Guibas, L. J. (2019), KPConv: Flexible and Deformable Convolution for Point Clouds, pp. 6411–6420.

Tran, H., Khoshelham, K., Kealy, A. and Díaz Vilariño, L. (2018), 'Shape Grammar Approach to 3D Modeling of Indoor Environments Using Point Clouds', *Journal of Computing in Civil Engineering* **33**.

Wang, B., Yin, C., Luo, H., Cheng, J. C. P. and Wang, Q. (2021), 'Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data', *Automation in Construction* **125**, 103615.

Zhang, G., Vela, P. A. and Brilakis, I. (2014), 'Automatic Generation of As-Built Geometric Civil Infrastructure Models from Point Cloud Data', pp. 406–413. ISBN: 9780784413616 Publisher: American Society of Civil Engineers.