Geometry updating for digital twins of buildings: a review to derive a new geometry-based object class hierarchy

Conference Paper · July 2022

DOI: 10.35490/EC3.2022.155

3 authors:

Zhiqi Hu
University of Cambridge

Yasmin Fathy
University of Cambridge

Ioannis Brilakis
University of Cambridge

GEOMETRY UPDATING FOR DIGITAL TWINS OF BUILDINGS: A REVIEW TO DERIVE A NEW GEOMETRY-BASED OBJECT CLASS HIERARCHY

Zhiqi Hu[1], Yasmin Fathy[1], Ioannis Brilakis[1]
[1]University of Cambridge, Cambridge, United Kingdom

Abstract

Geometry updating for digital twins of buildings is a time-consuming and manual task, resulting in poor progress monitoring and quality control during the construction stage. This paper reviews the state of the art in practice and research on spatial and visual data-based approaches for updating digital twin geometry of buildings. We draw novel key insights into the effectiveness, experiments, and limitations of seven classes of methods summarised from the most recent papers. Consequently, four core gaps in knowledge are investigated. Finally, a new geometry-based object class hierarchy is derived to support geometry updating for maintaining digital twins in future directions.

Introduction

A Building Digital Twin (BDT) serves as a digital representation of a physical building that mirrors the building's status and behaviour throughout its lifecycle from the design, construction to operation stages. Geometry updating of BDT can keep the product information of an asset up-to-date. A building refers to the structure comprised of connected object instances (e.g., walls, roofs, beams, columns, windows, doors, etc.) along with Mechanical, Electrical and Plumbing (MEP) systems (e.g., piping and duct systems, fire protection systems, etc.). Geometry updating refers to detecting building object instances from the on-site collected Spatial and Visual Data (SVD) at a given timestamp during the construction stage, then meshing the data and assigning it as a timestamped three-dimensional (3D) representation in the BDT. SVD refers to Point Cloud Data (PCD) as spatial data and images or video sequences as visual data that are acquired by terrestrial or mobile scanners and cameras. Additionally, the final as-designed file at the end of the design stage is marked as "Design Intent" (DI). It can assist the geometry updating of BDT during the construction stage and serve as a benchmark for evaluating the project performance.

One of the greatest challenges faced by the Architecture, Engineering, and Construction (AEC) industry is poor project performance due to the lack of timely progress monitoring and quality control during the construction stage. It is estimated that only around 34% of large projects are delivered on budget and only about 37% of projects are completed on time (Countryman et al., 2020). Over 50% of construction companies have long experienced dealing with under-performing projects (Armstrong, 2015). AEC industry can benefit significantly from BDT applications, including up to a 50% increase in productivity, 10% decline in schedules, and 80% reduction in rework (Fingland, 2019). Dynamically updating BDT geometry from SVD with the help of the DI is a core step for automating progress monitoring and quality control. This step will help provide the geometric discrepancy data between as-designed and as-built geometry to measure progress and evaluate spatial quality during the construction stage.

This paper aims to gain a contextual understanding of the state of practice in BDT geometry updating and investigate the state of research regarding building object instance detection in the environment of contrasting SVD-borne and DI-borne geometry (SVD-vs-DI). We analysed three stages of BDT geometry updating and the current building object classification standards to deliver the core knowledge gaps and a new geometry-based object class hierarchy as the main contribution.

State of Practice

The evaluation of state-of-the-art applications from the perspective of updating BDT geometry can help to understand their limitations and guide the literature review. Based on the Scan-vs-BIM system developed by Bosché et al. (2015), the state of practice here is discussed based on **three stages** of BDT geometry updating:

- DI geometry to SVD registration
- SVD-vs-DI object instance detection
- Object instance geometry capture and recording

The purpose of the first stage is to ensure that the DI file (e.g., IFC file) and the as-built SVD can be registered into a common coordinate system to facilitate the comparison between the DI-borne geometry and the as-built status during the construction stage. The second stage contains PCD-vs-DI object instance detection and image-based object instance detection. The output data of this stage can be used to capture 3D geometry of object instances. The final stage converts the PCD as low-level digitised 3D data into high-level information-rich 3D formats (e.g., 3D surface mesh) to support progress monitoring and quality control. The following discusses the state of practice of three stages mentioned above in detail:

The first stage is a commercially solved problem. The user needs to manually find at least three corresponding points both in the PCD and the DI geometry, then the software can automatically calculate the transition and rotation matrix to register the PCD against the DI. Another way to achieve PCD to DI geometry registration is to manually adjust origins and axes to make coordinate systems the same. Images can be registered by simulating the camera poses in the 3D DI geometry to capture 2D pictures. It requires the camera's intrinsic and extrinsic parameters.

As for the second stage, no commercial software can automatically detect as-built object instances and match them with as-designed models in the SVD-vs-DI environment. Some software can do clash detection between PCD and DI geometry, but they cannot focus on the instance level to detect and extract the whole individual component in the PCD. Figure 1 shows state-of-the-art software classification for clash detection. The experiment using Faro As-Built for Revit for ISPRS benchmark TUB1 input data (Khoshelham et al., 2017) with the upper range clearance 50 mm took 22 minutes to complete clash detection on the desktop (Processor: AMD Ryzen 5 5600X 6-Core Processor; RAM: 32GB; GPU: AMD Radeon RX 6800). The result of 87 clashed elements contains over 90% unnecessary collisions such as noisy points of a part of an object instance. Therefore, clash detection cannot be directly used to match object instances for BDT geometry updating.

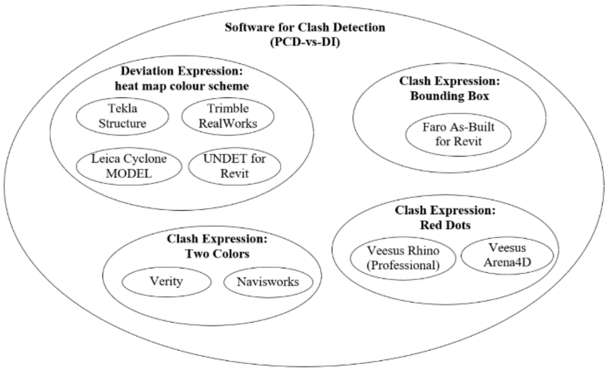

Figure 1: Software classification for clash detection

As for the third stage, no software can automatically capture and record the 3D geometry of as-built object instances from the PCD with the help of DI. By contrast, two commercial solutions named OpenSpace and Buildots can measure construction progress by capturing images with a hat-mounted 360° camera. The image data stream is then compared with the expected progress from the DI to update the progress situation. Buildots claims that it can also evaluate visual quality such as automatically detecting the wrong place of the as-built window in the image. Nevertheless, these two solutions only rely on visual inspections to detect quality related issues; they cannot update BDT geometry in 3D view. The updated 3D geometry is essential to evaluate spatial quality during construction. Meanwhile, some software can automatically extract or generate limited object classes only from the PCD without being guided by the DI. For example, EdgeWise can automatically detect and generate cylindrical pipe segments, round joints (e.g., elbows, reducers), and rectangular duct segments without any manual effort. However, it is designed for PCD-to-BIM rather than PCD-vs-DI. It cannot help to distinguish which as-built object instance belongs to which DI instance.

Besides the three stages discussed above, we also need to understand what object classes, as well as shape classes, exist in a typical building before updating its geometry in the BDT. Various classification systems have been developed by different nations and institutions during the last sixty years, such as Uniclass, UniFormat, and OmniClass (Afsari and Eastman, 2016). However, each standard has its own criteria, and all these existing classification systems are only function-oriented to support activities during the building's lifecycle. They cannot be applied directly to facilitate updating BDT geometry at the current stage.

Overall, to the best of our knowledge, there is no state-of-practice solution that can automatically keep BDT geometry updated based on the DI during the construction stage to support project management.

State of Research Overview

The first and third stages elaborated above have been well-solved in different studies in recent years. Random Sample Consensus (RANSAC)-based plane extraction (Bosché, 2012; Bueno et al., 2018) and line extraction (Stojanovic et al., 2018; Kaiser et al., 2019) are core ideas to (semi-)automatically achieve coarse registration in the first stage, along with Iterative Closest Point (ICP) algorithm and its variants (e.g., progressive ICP, Go-ICP, etc.) (Tang et al., 2013; Yang et al., 2013) to achieve fine registration for more precise results. For the third stage, 3D representation to convert the extracted points into high-level information-rich 3D formats has also been well-explored. Primitive shape-based methods (e.g., B-Rep fitting) perform well in representing primitives such as cuboid and cylinder but cannot describe in full details of irregular object instances. By contrast, meshing (Abdelkader et al., 2020; Cheng et al., 2008; Kazhdan et al., 2006) is effective to generate detailed representations to retain more geometric properties of objects. Other mesh-based variants (Hong et al., 2017; Groueix et al., 2018; Otoguro et al., 2018) have also been proposed recently to generate high-quality structured meshes for complex and deformable shapes. It should be noted that incomplete PCD (e.g., with holes or truncation) can lead to poor or wrong results of meshing. To this end, Rashidi and Brilakis (2016) summarised the methods for filling gaps in PCD to improve the performance of meshing.

The second stage for updating BDT geometry has more space for researchers to explore, and thus is the main part of this review. Table 1 summarised the findings of investigated papers. Overall, SVD-vs-DI object instance detection is split into **two categories** based on 3D/2D data formats: PCD-vs-DI instance extraction and image-based instance extraction. As for the first category, PCD-vs-DI instance detection is comprised of three workflows (Liu et al., 2021) depending on the input data formats:

- **Workflow 1:** comparing the as-built PCD with the as-designed PCD generated from the DI geometry.
- **Workflow 2:** comparing the as-built PCD directly with the DI geometry.

Table 1: Summary of the second stage for updating BDT geometry: SVD-vs-DI object instance detection

Categories	Workflows	Methods	Existing Studies	Experiments	Core Limitations
PCD-vs-DI instance detection	As-built PCD vs As-designed PCD	Point-to-point comparison	Bosché and Haas, 2008; Turkan et al., 2012; Turkan et al., 2013; Bosché et al., 2014; Turkan et al., 2014	column, beam, slab, wall, cylindrical pipe, rectangular duct, formwork, scaffolding, shoring, rebar	Fail to detect instances when the deviation of position > 50 mm
		Hough transform	Ahmed et al., 2014; Bosché et al., 2015	cylindrical pipe, round elbow	Complex computation; Fail to detect highly occluded objects
	As-built PCD vs DI geometry	Feature-based method	Kim et al., 2013; Kalasapudi et al., 2014; Gao et al., 2016	wall, ceiling, column, beam, slab, rectangular duct, cylindrical pipe, round elbow/reducer	As-built must be the same as as-designed; Require as-built without any occlusion and noise; Fail to detect glass-made or curved planes
		Point-to-surface comparison	Zhang and Arditi, 2013; Gao et al., 2016; Tran and Khoshelham, 2019	column, wall, cylindrical pipe, round elbow, round reducer, rectangular duct	Cannot extract all points corresponding to the object instance
		RANSAC	Kim et al., 2016; Nguyen and Choi, 2018; Guo et al., 2020; Rausch and Haas, 2021	precast slab, wall, cast-in-place footing, cylindrical pipe, duct, cable tray	Only robust for primitive shapes
	As-built mesh vs DI geometry	Mesh-supported method	Date et al., 2012; Kim et al., 2020	cylindrical pipe, U-shape round joint, wye joint, cross joint, slab, wall, beam, column	Loss of Point information; Small and highly occluded instances may be missed
Image-based instance detection	Deep learning	CNN & variants	Czerniawski et al., 2020; Kufuor et al., 2021; Ying et al., 2019; Hou et al., 2020	window, stairs, wall, elevator, duct, column, beam, slab, door, pipe, socket, switch, radiator	Lack 3D data to reflect spatial information

- **Workflow 3:** comparing the as-built mesh generated from the as-built PCD with the DI geometry.

Workflow 1 requires generating the as-designed PCD from the DI geometry before detecting object instances in the as-built PCD. Each as-designed point can be calculated by projecting a scanning ray on the STL-format geometry (Bosché et al., 2014). Then, point-to-point comparison and Hough transform are two main methods used in this workflow for instance extraction. Workflow 2 directly uses the DI geometry format (e.g., IFC, CAD, STL, etc.) to support instance extraction from the as-built PCD. The advantage is that it keeps more initial features of the DI geometry than Workflow 1. Feature-based methods, point-to-surface comparison, and RANSAC are three core methods in this workflow. Lastly, workflow 3 requires generating meshes from the as-built PCD before detecting object instances with the DI geometry prior.

Image-based instance detection as the second category is another way for construction progress monitoring and quality control. We tend to investigate the advanced methods (**workflow 4:** deep learning) in this field since 2D images can be considered as auxiliary input data to support geometry updating. The next two sections will elaborate on the seven core methods from four workflows with experiments and limitations summarised in table 1.

PCD-vs-DI Instance Detection

Point-to-point comparison was first used to automatically retrieve 3D object instances in the as-built PCD by Bosché and Haas (2008). The retrieval rate $R\%$ is calculated by the ratio of the number of retrieved as-designed points to the total number of as-designed points. The threshold is set as 50% to assess the retrieval result. Initial experiments on small-scale datasets (4 columns and 1 slab, each within 18,000 points) presented a robust result for the proof of concept. This method has then been applied to detect primary structural object classes (Turkan et al., 2012; Turkan et al., 2013), to track secondary and temporary structural object instances (Turkan et al., 2014) and to detect mechanical object classes (Bosché et al., 2014) for progress monitoring at the construction stage. The method performed well in tracking the status of structural instances but produced high rates of false negative and false positive results when detecting mechanical instances with large spatial deviations (out of 50 mm) against the DI. This problem cannot be avoided even by adding the surface normal vector as an additional rule to support instance detection.

Hough transform performs well in shape detection in the complex environment with noise. It was first developed for line detection in a cluster of 2D noisy points (Duan et al., 2010). An edge point (x_i, y_i) on the line in the image space can be transformed in the parameter space. The edge line can then be detected if the corresponding lines in the parameter space cross the same point. A 2D Hough transform-based object instance detection method has been developed to extract cylindrical piping segments (Ahmed et al., 2014). A cluster of 3D point slices needs to be projected along with the estimated object normal orientation from the DI geometry before the 2D circle slices are determined by Hough transform. Then, the circle slices will be integrated to grow cylindrical pipe segments. Nevertheless, this method requires that the as-built position and dimension of the object instance are the same as the DI geometry. Bosché et al. (2015) were inspired from this method and combined the point-to-point comparison and Hough transform together to detect cylindrical MEP components. It overcomes the limitation from Bosché et al. (2014) on detecting out-of-place instances (within 2 meters) and can identify the instance completeness through detection. However, Hough transform is computationally complex. The method does not consider the effect of high occlusions and clutter (e.g., stuff in front of the instance) in the PCD. It assumes that the most cylindrical instances are built in the orthogonal direction, which leads to the methods with less robustness in complex environments.

Feature-based method uses object features (e.g., position, scale, colour, etc.) to detect instances. A three feature-based (Lalonde feature, orientation, and continuity) instance detection method has been developed to detect linear and surface instances to measure construction progress (Kim et al., 2013). This method is robust in the noisy environment but assumes that all object instances are constructed according to the DI. Similarly, a five feature-based method has been proposed to match object instances. The features include length, size, colour, orientation, and the number of connections with adjacent instances (Kalasapudi et al., 2014). This method was only tested for prefabricated pipe detection in an environment without any occlusion and clutter. A distribution-based method has been developed by computing the probability distribution of the geometric properties for both PCD and the DI file (Gao et al., 2016). However, this method requires the denoised PCD without any occlusion. All these feature-based methods cannot deal with the detection of glass-made object instances or curved planes.

Point-to-surface comparison calculates the ratio of the overlapping area between the extracted points and the object instance from the DI (Zhang and Arditi, 2013; Gao et al., 2016). Tran and Khoshelham (2019) developed the surface coverage ratio calculation algorithm by using alpha shape reconstructed from the orthogonal projection of points to make the method more robust. The method has been used in the PCD-vs-DI instance detection including columns, walls, duct, and piping segments. However, this kind of method cannot extract all points corresponding to the instance when there are deviations between the PCD and DI geometry or in the PCD with high clutter. Also, the coverage ratio threshold needs to be manually set. The method will be invalid if the deviation of the as-built position or orientation exceeds this ratio threshold.

RANSAC is more robust than all methods discussed above for detecting instances from the as-built PCD with over 50% of outliers based on the DI prior (Schnabel et al., 2007). RANSAC can detect and extract geometric primitives including planes, spheres, and cylinders from PCD. It has been applied to optimise the edge points extracted from as-built PCD to assess the quality of the precast slabs (Kim et al., 2016). It has also been employed with the normal-based region growing method and K-Nearest Neighbours (KNN) to detect cylindrical pipe segments when the position and orientation of as-built instances differ from the DI (maximum orientation error 7.5°; maximum position error 35 mm) (Nguyen and Choi, 2018). Similarly, Guo et al. (2020) used RANSAC to detect cuboid-shape instances and a variant of RANSAC named Maximum Likelihood Estimation Sample Consensus (MLESAC) to fit cylinder-shape pipes. Rausch and Haas (2021) also applied RANSAC for cast-in-place footing detection. However, all RANSAC-based methods can only detect primitive-shape object instances such as cylindrical pipe segments (cylinders), rectangular duct segments (cuboids), and floors (planes). The object instances with moderate or complex shapes such as T-shape pipe joints, cross-shape duct joints, heating terminals, and sprinklers cannot be detected directly. Also, the lack of a checking step may lead to false negative or false positive results when detecting instances with high clutter and occlusions.

Mesh-supported method requires generating meshes from the as-built PCD before detecting instances with the DI prior. Spin image has been developed for 3D object instance detection from the mesh (Date et al., 2012). It is a data-level shape descriptor representing the surface of the instance by bilinear interpolation. The corresponding points in both as-built and as-designed meshes can be matched by comparing spin images. Kim et al. (2020) also generated a mesh from the sparse PCD to semi-automatically detect instances for quantity calculation and progress monitoring at the construction stage. However, mesh generation requires downsampling PCD, resulting in a loss of information contained in the PCD. Spin image matching only investigates 20% to 50% of vertices, which means that small or highly occluded instances may be missed. Investigating all vertices may avoid this problem but leads to high computational complexity.

Image-based Instance Detection

Deep Learning (DL) is more competitive than the traditional machine learning pipeline to detect object instances from images in complex environments. The typical architecture of a Convolutional Neural Network (CNN) in DL is made up of three types of layers: the convolutional layer to abstract a feature map from the input image; the pooling layer to reduce the spatial size of convolved features and extract dominant features, and the fully connected layer to produce an output vector for classification (Zhao et al., 2019).

Many methods based on well-designed CNN architecture have been developed to achieve building object instance detection from images. A method based on DeepLab (Chen et al., 2017) has been proposed to automatically segment RGB-D images into 13 building object classes (window, floor, stairs, wall, etc.) with 0.50 IoU (Czerniawski et al., 2020). The Faster R-CNN (Girshick, 2015) uses the convolutional network to directly generate candidate regions. It has been applied to detect building electrical instances by training both RGB 360° and standard images (Kufuor et al., 2021). However, this method can only locate the instance position with bounding box. Mask R-CNN is then used to determine the boundary of instances in images (Ying et al., 2019). Overall, R-CNN-based detection methods can achieve high accuracy (over 90%) but real-time performance is deficient. To this end, a deeply supervised object detector (DSOD) combing Faster R-CNN and YOLO (Redmon et al., 2016) has been developed to detect structural object instances in the real-time scenario (Hou et al., 2020). This method has higher detection precision and recall (average 95%) and can be used to detect multiple instances in the complex construction environment, but it only performs well in detecting instances with primitive shapes.

Discussion

Four core gaps in knowledge have been identified based on the literature review regarding as-built object instance detection in the SVD-vs-DI environment. We do not yet know how to update BDT geometry in the following cases:

- environments with high clutter and occlusions. For example, only a part of an instance's surface is visible and captured by the scanner. Besides, other existing stuff can also cause occlusions during scanning.
- when there are distinct deviations in terms of position, orientation, scale, and shape between the DI geometry and the as-built instances (e.g., axis deviation over 50 mm; angle deviation over 15°).
- when the instances built in non-primitive shapes. Cylinder and cuboid are the most prevalent primitive shapes used in construction. By contrast, other object types such as pipe joint, sprinkler, and light fixture are built in complex shapes, which are rarely detected by current methods.
- when the instance is transparent. In such a case, it is difficult to represent glass-made windows by point clouds due to the light transmission.

Meanwhile, the number of object classes in buildings can measure in the thousands, while most of them are rarely used. Object instance detection prioritising the top frequent object classes can significantly save time and reduce the cost of the BDT geometry updating, thus can serve progress monitoring and quality control better. Also, we previously concluded that the current object classification standards cannot be applied directly to support BDT geometry updating in the section of the state of practice. Therefore, we developed a new geometry-based building object class hierarchy from the geometric perspective named Hudrokis Tree (H.T.) (Figure 2). The Hudrokis Tree is comprised of three categories from the building composition's perspective: structural (S.), mechanical (M.), and electrical (E.). The structural category contains 4 primary object classes and 10 enriched object classes. The mechanical category includes 18 object classes from the plumbing, heating, and air conditioning systems and 4 object classes from the fire protection system. Similarly, the electrical category includes 12 object classes from the electrical supply, 3 object classes from the transport system, and 4 object classes from the control system. The last two layers (leaves) guide the most possible geometric shapes for each object class which can be used in the geometry updating work.

The Hudrokis Tree meets two core requirements that make it more applicable than existing classification standards: 1) It contains all common building object types of interest to the design, construction, and operation stages. Any non-visible object types such as foundations, piles, and ground beams are not included in the Hudrokis Tree since they are out of scope given that they are not visible in SVD. 2) It is a shape-oriented classification hierarchy. Also, this tree merged some object types with different functions but the same shape class into one object class. For example, we merged duct, piping, and drain segments into one segment class named "plumbing segment" since all these object types only have two geometry classes in practice: cylinder and cuboid. Overall, Hudrokis Tree is the prerequisite for ranking the top frequent object classes in a typical building and is considered as a general guide employed for the BDT geometry generating and updating in the future.

Conclusion

Buildings are not static; BDT endows them with dynamic characteristics. Structural, mechanical, and electrical components work together to build product information for BDT through the design, construction, operation and renovation stages. Keeping BDT dynamic by updating its geometry can facilitate progress monitoring and quality control, and thus support project management. PCD is widely used to extract 3D information in BDT geometry updating in decade years, but the main gaps mentioned before still need to be solved. Image-based methods are more capable of progress monitoring rather than quality control since it lacks 3D semantic information of as-built object instances. Combining PCD with images to develop hybrid methods can be one of the future directions to overcome challenges. Employing and developing machine learning and CNN-based methods can fasten the automation level of updating. Meanwhile, Hudrokis Tree can be used as a classification guide to facilitate generating and updating BDT geometry at any building lifecycle stages.

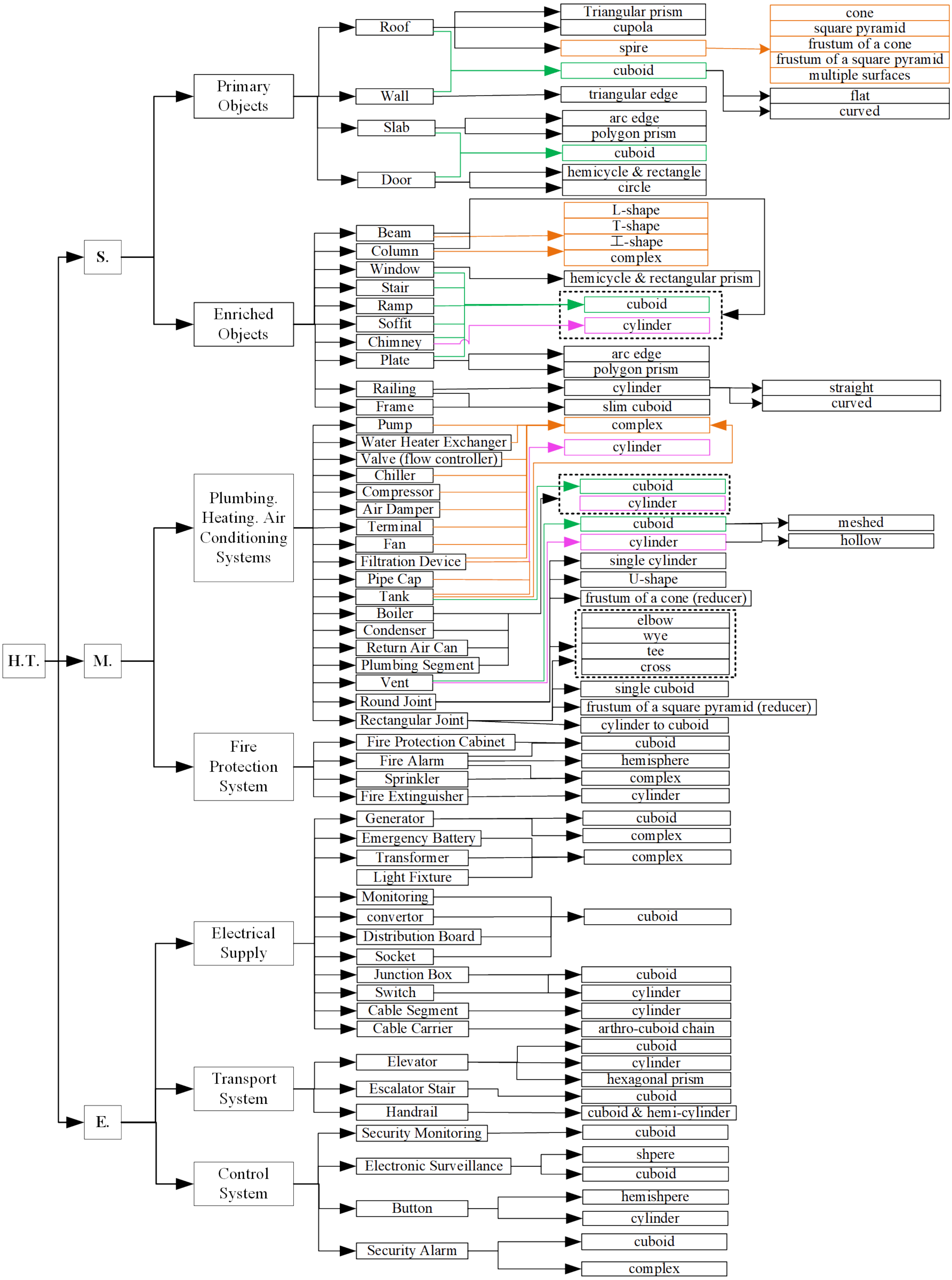

Figure 2: New geometry-based building object class hierarchy – Hudrokis Tree (H.T.)

Acknowledgments

This work is funded by European Commission's Horizon 2020 for 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. The authors acknowledge the help of Viktor Drobnyi in revising Hudrokis Tree.

References

Abdelkader, A., Bajaj, C. L., Ebeida, M. S., Mahmoud, A. H., Mitchell, S. A., Owens, J. D., & Rushdi, A. A. (2020). VoroCrust: Voronoi meshing without clipping. ACM Transactions on Graphics (TOG), 39(3), 1-16.

Afsari, K., & Eastman, C.M. (2016, April). A comparison of construction classification systems used for classifying building product models. In 52nd ASC Annual International Conference Proceedings (pp.1-8).

Ahmed, M. F., Haas, C. T., & Haas, R. (2014). Automatic detection of cylindrical objects in built facilities. Journal of Computing in Civil Engineering, 28(3), 04014009.

Armstrong, G. (2015). Global Construction Survey 2015: Climbing the curve. KPMG International.

Bosché, F.(2012).Plane-based registration of construction laser scans with 3D/4D building models. Advanced Engineering Informatics,26(1),90-102.

Bosche, F., & Haas, C. T. (2008). Automated retrieval of 3D CAD model objects in construction range images. Automation in Construction, 17(4), 499-512.

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., & Haas, R. (2015). The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. Automation in Construction, 49, 201-213.

Bosché, F., Guillemet, A., Turkan, Y., Haas, C. T., & Haas, R. (2014). Tracking the built status of MEP works: Assessing the value of a Scan-vs-BIM system. Journal of computing in civil engineering, 28(4), 05014004.

Bueno, M., Bosché, F., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2018). 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. Automation in Construction, 89, 120-134.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.

Cheng, S. W., Dey, T. K., & Levine, J. A. (2008). A practical Delaunay meshing algorithm for a large class of domains. In Proceedings of the 16th International Meshing Roundtable (pp. 477-494). Springer, Berlin, Heidelberg.

Countryman, T., Narsalay, R., Webster, A., Bajla, P., & Pohle, F. (2020). Building More Value With Capital Projects. Accenture.

Czerniawski, T., & Leite, F. (2020). Automated segmentation of RGB-D images into a comprehensive set of building components using deep learning. Advanced Engineering Informatics, 45, 101131.

Date, H., Kaneta, Y., Hatsukaiwa, A., Onosato, M., & Kanai, S. (2012). Object recognition in terrestrial laser scan data using spin images. Computer-Aided Design and Applications, 9(2), 187-197.

Duan, D., Xie, M., Mo, Q., Han, Z., & Wan, Y. (2010, October). An improved Hough transform for line detection. In 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) (Vol. 2, pp. V2-354). IEEE.

Fingland, J. (2019, March 18). Preparing for Digital Transformation in the AEC Industry. Trimble.

Gao, T., Ergan, S., Akinci, B., & Garrett, J. (2016). Evaluation of different features for matching point clouds to building information models. Journal of Computing in Civil Engineering, 30(1), 04014107.

Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). A papier-mâché approach to learning 3d surface generation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 216-224).

Guo, J., Wang, Q., & Park, J. H. (2020). Geometric quality inspection of prefabricated MEP modules with 3D laser scanning. Automation in Construction, 111, 103053.

Hong, Y. M., Zhang, J. H., Chen, C. Y., Wünsche, B. C., Chien, H. J., & Ying, T. K. (2017, December). Surface reconstruction of 3D objects using local moving least squares and KD trees. In 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ) (pp. 1-6). IEEE.

Hou, X., Zeng, Y., & Xue, J. (2020). Detecting structural components of building engineering based on deep-learning method. Journal of Construction Engineering and Management, 146(2), 040190.

Kaiser, T., Clemen, C., & Maas, H. G. (2019). Automated Alignment of Local Point Clouds in Digital Building Models. International Archives of the

Photogrammetry, Remote Sensing and Spatial Information Sciences, 42(5/W2).

Kalasapudi, V. S., Turkan, Y., & Tang, P. (2014, December). Toward automated spatial change analysis of MEP components using 3D point clouds and as-designed BIM models. In 2014 2nd International Conference on 3D Vision (Vol. 2, pp. 145-152). IEEE.

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006, June). Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing (Vol. 7).

Khoshelham, K., Vilariño, L. D., Peter, M., Kang, Z., & Acharya, D. (2017). The ISPRS benchmark on indoor modelling. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 42, 367-372.

Kim, C., Son, H., & Kim, C. (2013). Automated construction progress measurement using a 4D building information model and 3D data. Automation in construction, 31, 75-82.

Kim, M. K., Wang, Q., Park, J. W., Cheng, J. C., Sohn, H., & Chang, C. C. (2016). Automated dimensional quality assurance of full-scale precast concrete elements using laser scanning and BIM. Automation in Construction, 72, 102-114.

Kim, S., Kim, S., & Lee, D. E. (2020). 3D point cloud and BIM-based reconstruction for evaluation of project by as-planned and as-built. Remote Sensing, 12(9), 1457.

Kufuor, J., Mohanty, D. D., Valero, E., & Bosché, F. (2021, January). Automatic MEP Component Detection with Deep Learning. In International Conference on Pattern Recognition (pp. 373-388). Springer, Cham.

Liu, J., Xu, D., Hyyppa, J., & Liang, Y. (2021). A survey of applications with combined BIM and 3D laser scanning in the life cycle of buildings. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Nguyen, C. H. P., & Choi, Y. (2018). Comparison of point cloud data and 3D CAD data for on-site dimensional inspection of industrial plant piping systems. Automation in Construction, 91, 44-52.

Otoguro, Y., Takizawa, K., & Tezduyar, T. E. (2018). A general-purpose NURBS mesh generation method for complex geometries. In Frontiers in Computational Fluid-Structure Interaction and Flow Simulation (pp. 399-434). Birkhäuser, Cham.

Rashidi, A., & Brilakis, I. (2016). Point cloud data cleaning and refining for 3D as-built modeling of built infrastructure. In Construction Research Congress 2016 (pp. 919-929).

Rausch, C., & Haas, C. (2021). Automated shape and pose updating of building information model elements from 3D point clouds. Automation in Construction, 124, 103561.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Schnabel, R., Wahl, R., & Klein, R. (2007, June). Efficient RANSAC for point-cloud shape detection. In Computer graphics forum (Vol. 26, No. 2, pp. 214-226). Oxford, UK: Blackwell Publishing Ltd.

Stojanovic, V., Richter, R., Döllner, J., & Trapp, M. (2018). Comparative visualization of BIM geometry and corresponding point clouds. Building Information Systems in the Construction Industry, 13.

Tang, P., & Rasheed, S. H. (2013, December). Simulation for characterizing a progressive registration algorithm aligning as-built 3D point clouds against as-designed models. In 2013 Winter Simulations Conference (WSC) (pp. 3169-3180). IEEE.

Tran, H., & Khoshelham, K. (2019). BUILDING CHANGE DETECTION THROUGH COMPARISON OF A LIDAR SCAN WITH A BUILDING INFORMATION MODEL. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences.

Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012). Automated progress tracking using 4D schedule and 3D sensing technologies. Automation in construction, 22, 414-421.

Turkan, Y., Bosché, F., Haas, C. T., & Haas, R. (2013). Toward automated earned value tracking using 3D imaging tools. Journal of construction engineering and management, 139(4), 423-433.

Turkan, Y., Bosché, F., Haas, C. T., & Haas, R. (2014). Tracking of secondary and temporary objects in structural concrete work. Construction Innovation.

Yang, J., Li, H., & Jia, Y. (2013). Go-icp: Solving 3d registration efficiently and globally optimally. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1457-1464).

Ying, H. Q., & Lee, S. (2019). A mask r-cnn based approach to automatically construct as-is ifc bim objects from digital images. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 36, pp. 764-771). IAARC Publications.

Zhang, C., & Arditi, D. (2013). Automated progress control using laser scanning technology. Automation in construction, 36, 108-116.

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.