Planes geometry verification program written in Python

Aset Madiev*, *Jan* Erdelyi[1], and *Richard* Honti[1]

[1] Department of Surveying, Faculty of Civil Engineering, Slovak University of Technology in Bratislava, Vazovova 5, 812 43, Bratislava, Slovakia

Abstract. In recent years, there has been significant progress in sensor and scanning technologies in the computer vision and 3D visualization field. This has led to an increased interest in processing and analysing point clouds, which are groups of 3D points collected by laser scanners, stereo cameras, and drones. The data obtained from these devices has a wide range of applications. This article focuses on using point cloud data to verify planar structural elements of buildings. The article aims to present a methodology for analysing planar point cloud objects using 3D models in IFC format as reference data. The research includes processing data in IFC format, exporting geometric information, segmenting plane objects from a point cloud obtained by laser scanning, and analysing the data obtained. We provide the necessary graphic materials and code to enhance the content of the article. We also evaluate the results and create an outline for future work.

1 Introduction

In recent decades, the incorporation of automation within the construction sector has significantly contributed to heightened productivity and operational efficiency. The evolution of spatial data acquisition methodologies, specifically photogrammetry and laser scanning, has ushered in novel prospects for innovative applications in the realm of construction [1].

The research, as delineated in [2], delves into the practicability of employing real-time photogrammetric techniques for deformation measurements. Concurrently, [3] introduces an inventive algorithm designed to autonomously detect spheres and derive their parameters from three-dimensional point clouds.

Nonetheless, a burgeoning sector in contemporary times is the adoption of Building Information Modeling (BIM) for the automated scrutiny of architectural structures. A pivotal paradigm within this domain is the scan-vs-BIM concept, entailing the fusion of BIM and point clouds to assess construction quality. This conceptual framework entails scanning the subject structure, generating a point cloud from the resultant data, inclusive of point cloud registration and data filtration. Subsequently, discrepancies (distances) between the envisioned model and the point cloud are ascertained based on the acquired information.

---

*Corresponding author: aset.madiev@stuba.sk

The present investigation endeavors to devise an automated system for assessing the geometry of planar structural elements, employing software developed in the Python programming language. BIM serves as the principal reservoir for extracting geometric data.

This paper focuses on formulating and deploying a software solution for automated verification of planar structural elements geometry. The methodology encompasses the manipulation and extraction of geometric data from BIM models presented in the Industry Foundation Classes (IFC) format. The pivotal stages of the study encompass the segmentation of plane structures based on point clouds acquired through Terrestrial Laser Scanning (TLS), followed by an analysis and visualization of any identified deviations.

The objective of the research is not only to furnish a technical rationale for the methodology but also to proffer practical recommendations for refining the process of validating building structure geometry. The concluding sections of this work will present deductions grounded in the research outcomes and delineate future initiatives aimed at enhancing methodologies for validating building structure geometry utilizing point cloud data and BIM technology.

2 Materials and methods

The section expounds upon the exhaustive methodology and approaches employed in the investigation. The elucidation encompasses the intricacies of extracting geometric data from Industry Foundation Classes (IFC), expounding on the segmentation technique applied to plane structures utilizing data obtained through Terrestrial Laser Scanning (TLS). Emphasis is directed towards the deployment of a software solution designed to automate the validation of plane geometric data through the utilization of the Python programming language.

2.1 Extraction data from IFC

The initial phase of the investigation encompassed the retrieval of information from Industry Foundation Classes (IFC). The selection of IFC was predicated on its widespread adoption owing to its open specification facilitating data exchange within the realms of construction and facility management. The progenitor of this format is the BuildingSMART alliance, previously denominated as the International Alliance of Interaction (IAI). At the crux of this alliance lies the tenet of open BIM, signifying the overarching principle of transparency and standardization of data accessibility for all stakeholders involved in a project, irrespective of the software employed for task resolution. Presently, the latest iteration of IFC stands at IFC 4.3, with ongoing developmental efforts toward IFC 4.4 - dev [4].

Within the IFC framework, the structural attributes of a wall are delineated through entities such as IfcWall or IfcWallStandardCase. To extract geometric details, it is imperative to ascertain the origin coordinates and orientation of the local coordinate system to the World Coordinate System (WCS) for the specified wall. The IfcLocalPlacement, featuring attributes RelativePlacement and PlacementRelTo, delineates the position of the coordinate system's origin. In instances where PlacementRelTo is omitted, the wall's position is established within the WCS. The origin of the local coordinate system is discerned through IfcAxis2Placement3D, accessible via IfcLocalPlacement and employing IfcCartesianPoint.

The directional alignment of the X and Z axes is defined by IfcDirection, while the Y axis contributes to the formation of a right-handed Cartesian system. Subsequently, the elucidation of information about the wall's shape, as defined by IfcProductDefinitionShape, becomes imperative. For instance, if the wall assumes the form of a swept solid, the vertices of a 2D polygon delineating the wall's outline can be ascertained through IfcArbitraryClosedProfileDef. Sequential access to this entity involves traversing from IfcProductDefinitionShape to IfcShapeRepresentation and culminating in

IfcExtrudedAreaSolid. The latter entity additionally specifies the direction and extent of the wall's height (Fig. 1).

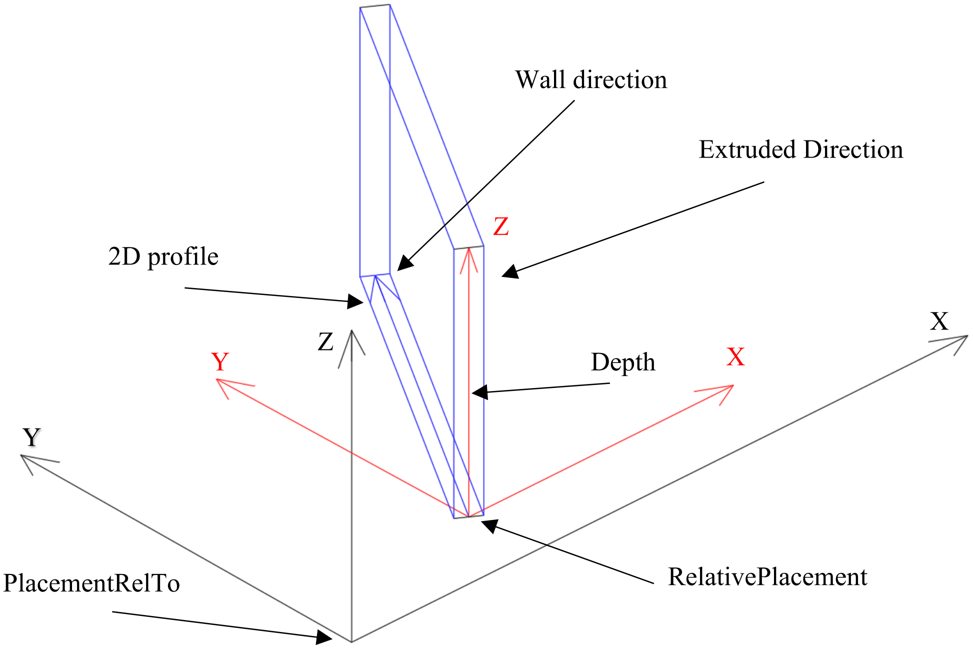

Fig. 1. Visualization of wall definition in IFC.

2.2 Plane's structure segmentation

The methodology for partitioning plane structures integrates a modified RANdom SAmple Consensus (RANSAC) approach with the region-growing method, as expounded in [3]. The segmentation procedure initiates by computing the normal vectors to the surface at selected points within the point cloud. Surface normal constitute crucial attributes of a geometric surface and find widespread utility in resolving diverse practical challenges, primarily linked to graphical representation. In this context, the attribute of normality facilitated the recognition of objects and surfaces within the point cloud. The intricacy of ascertaining the normal at a surface point is approximated by the challenge of estimating the normal of the plane tangent to the surface. Thus, the function responsible for normal computation for each point is contingent upon two pivotal parameters: the search radius and the maximum value of neighboring points.

Subsequently, the segmentation process proceeds from the initial point, defined as the point nearest to the center of gravity of the proposed model. This computation is based on the four corner points derived during the exportation of geometric data from Industry Foundation Classes (IFC). The most suitable regression plane is determined through the utilization of the k nearest points, aiming to minimize their orthogonal distances from the resultant plane via orthogonal regression. The value of k is automatically determined based on the local density of points surrounding the selected point. The optimal-fit regression plane is then computed using the k-nearest points, minimizing their orthogonal distances from the resultant plane through orthogonal regression. The determination of k is automated, relying on the local point density around the selected point.

Following this, the estimated regression plane undergoes scrutiny for nearest neighbors, and inlier points for this plane are identified. The selection of such points adheres to two criteria. Firstly, inlier points are those positioned closer to the plane than a specified threshold distance. Secondly, they are identified based on the orientation of the normal vector. Specifically, points are considered inliers if the angle between the local normal vector (at the point) and the normal vector of the resultant regression plane is less than a chosen threshold.

The identification of inlier points and the reevaluation of plane parameters occur iteratively by inspecting neighboring points and gradually increasing their number. This reevaluation process persists until all points pertaining to a specific plane are selected. The outcomes of the plane segmentation comprise subsets of point clouds correlated with the identified planes, accompanied by parameters of regression models (normal vector parameters, number of inlier points, standard deviation of the plane estimate) [5].

2.3 Software solution design

During this phase of the investigation, we developed and rigorously examined a specialized software application written in the Python programming language. The software's purpose is to automate the process of validating the geometry of plane structures. It comprises several critical modules that ensure a comprehensive data analysis and verification cycle. The initial stage of data processing from the IFC involves automated extraction and organization of information. This initial step involves retrieving data and identifying all walls within the IFC file structure. Each identified object is presented as a geometric “shape” object to enhance accessibility. After identifying walls, the next stage involves locating the vertices of each wall and converting its coordinates. Given the complexity of IFC files, coordinates primarily utilize a local coordinate system. Transformation matrices are used to convert the local coordinates of wall vertices into global coordinates that incorporate information about the wall’s spatial position and orientation. Additionally, to eliminate extraneous points within walls, such as connection points between two walls, all possible combinations of four corner points that form a plane along the length of the wall are generated. The centers of gravity of these planes are then compared to the overall wall's center of gravity. The points and their corresponding corner points are preserved if the difference is equal to half the width of the wall. Consequently, this algorithm yields the centers of gravity and their respective corner points in the global coordinate system. The subsequent module dedicated to wall segmentation processes laser scanning data and performs wall segmentation. First, a point cloud is loaded and normal are calculated for each point. A central point provided by the preceding module is used as the seed point for region growing for the segmentation process. The module then searches for the nearest neighboring points using a K-dimensional (KD) tree. In the initial iteration, the closest neighboring points are limited to twenty. Subsequently, these neighboring points are subject to the singular value decomposition (SVD) method. From this, plane normal and equation parameters are derived. Filtering is performed based on distance and plane inclination angle. Iterative refinement of the plane is then carried out by increasing the number of neighboring points until the growth ceases. The validation of the plane involves two stages. If the point count falls below a specified threshold or if the point cloud density is less than 80%, the plane is deemed invalid. Validated points are removed from the initial point cloud, restarting the process from a new seed point. The data comparison and visualization module facilitate a comparison between information from the IFC file and information acquired by TLS. Two deviation maps are created: the first illustrates the differences between the planned (IFC) and built (TLS) models. The second map depicts differences in wall flatness, calculated based on orthogonal distances to a regression plane. Several key methods were employed to evaluate the developed application's efficacy. A comparative analysis was conducted between automated and manual verification to assess method accuracy. Furthermore, the sensitivity of the system to variations in building information modelling (BIM) models and laser scanning data was evaluated through various input scenarios. Finally, the testing phase aimed to optimize the application's performance by considering enhancements such as parallel data processing and improved algorithms. During this phase, comprehensive testing and evaluation were conducted to assess the effectiveness of the developed system, as well as to identify areas for further optimization.

3 Results and discussion

An apartment in a residential building was scanned to showcase that the suggested procedure is viable. The property's name is Zwirn 1, and it is situated in Bratislava. The scanning process was carried out using a Trimble TX5 laser scanner. The scanning parameters were set as follows: field of view 360° in the horizontal direction and 305° in the vertical direction (from +90° to -62.5°), resolution 1/16, which means 24.5 mm *x* 24.5 mm at 10 m, 3x repetition of the measured distances. The a priori distance measurement error defined by the producer is ± 2 mm at 10 m. The apartment of the building was scanned from 13 positions of the scanner. The TLS data were aligned with the BIM model using surface-based registration (to join the individual scans) and target-based registration (to transform the point cloud into the coordinate system of the building). The registration error was 7 mm. The BIM model of the scanned apartment is shown in Fig. 2.

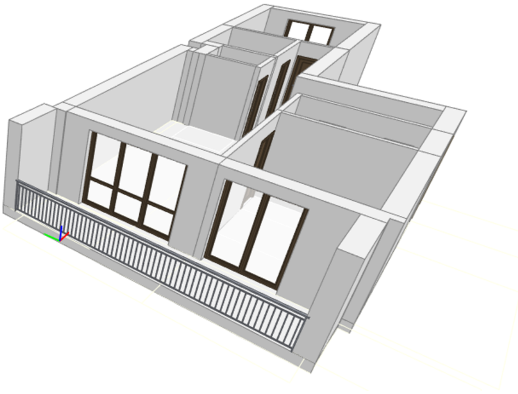

Fig. 2. The IFC model of the scanned apartment.

During the data processing phase, the following procedure involved the separation of planes from the entire point cloud stream. This was followed by a comparison with data derived from the IFC. To ensure accuracy in the results, visualization techniques were used at each stage of the algorithm. Subsequently, the next step involved visually representing the difference between the As-planned (from the building design) model and the As-built wall captured through TLS, as it is shown in Fig. 3.

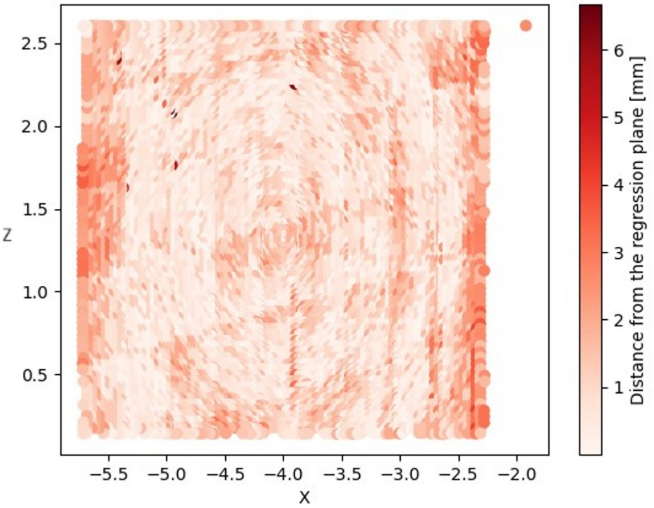

Fig. 3. Deviation (distance) between the as-planned (IFC) and the as-built (TLS) walls.

Figure 4 shows the deviation in the flatness of the wall, where orthogonal distances are calculated from the regression plane.

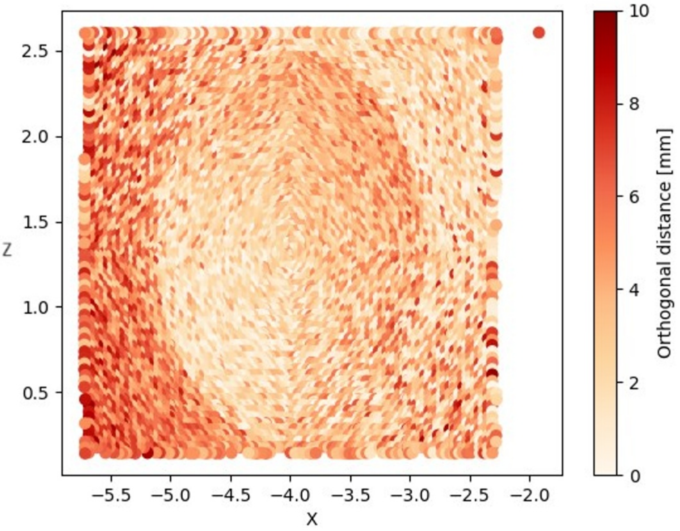

Fig. 4. Deviation map for the wall flatness quantification.

In most cases, the software solution has been proven to be efficient and effective. As part of our plans, we have an intention to integrate a graphical user interface for the program and expand the application's functionality to include additional structural elements.

4 Conclusion

This paper presents a methodology for automating the verification of plane structure geometries within the construction industry. The development of modern spatial data collection techniques, such as laser scanning and photogrammetry, has significantly enhanced the efficiency and streamlined processes within the building industry. Currently, significant emphasis is placed on processing and reviewing point clouds, which are three-dimensional arrangements of points collected through various methods.

The focus of this article is on the validation of building geometry using point clouds, as well as the extraction of geometric data from IFC. A crucial aspect involves the development of a software solution using the Python programming language to automate the validation process. This methodology involves processing and extracting geometric data from IFC files, segmenting buildings based on TLS data, and analyzing and visualizing deviations.

The findings from this research demonstrate the effectiveness of the proposed system, provide conclusions based on the achieved results, and outline future endeavors aiming to further enhance methodologies for validating the geometry of building structures using point clouds and BIM technology.

This publication was created with the support of the Scientific Grant Agency of the Ministry of Education, science, research and sport of the Slovak Republic and the Slovak Academy of Sciences for the project VEGA-1/0272/22.

This publication was created with the support of the Government office of the Slovak Republic from the funds of the Recovery and resilience plan of SR for the project no. 09I03-03-V01-00053

References

1. J. Peroš, R. Paar, V. Divić, and B. Kovačić, “Fusion of laser scans and image data - RGB+D for structural health monitoring of engineering structures, *Applied Sciences*, vol. 12, no. 22, 2021, Art no. 11763.
2. Ľ. Kovanič, P. Blistan, R. Urban, M. Štroner, K. Pukanská, K, Bartoš, and J. Palková, “Analytical determination of geometric parameters of the rotary Kiln by Novel approach of TLS point cloud segmentation,” *Applied Sciences*, vol. 10, no. 21, 2019, Art no. 7652.
3. R. Honti, J. Erdélyi and A. Kopáčik “A. semi-automated segmentation of geometric shapes from point clouds” *Remote Sensing,* vol. 14, no. 18, 2022.
4. buildingsSMART, 2018. [Online]. Available : http://www.buildingsmart-tech.org. Accessed: March 06, 2024.
5. V. Lehtola, S. Nikoohemat, A. Nüchter, “Indoor 3D: Overview on scanning and reconstruction methods,” *Handbook of Big Geospatial Data*, Springer, 2021, pp. 55–97.
6. F. Poux and R. Billen, “Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featuring vs. deep learning methods,” *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, 2019, Art no. 213.