sciendo

Slovak Journal of Civil Engineering

Vol. 28, 2020, No. 3, 13 – 19

DOI: 10.2478/sjce-2020-0019

AUTOMATED VERIFICATION OF BUILDING COMPONENTS USING BIM MODELS AND POINT CLOUDS

Richard, HONTI[1*], Ján, ERDÉLYI[1], Gabriela, BARICZOVÁ[1], Tomáš, FUNTÍK[2], Pavol, MAYER[2]

Abstract

One of the most important parts of construction work is the verification of the geometry of the parts of structures and buildings constructed. Today this procedure is often semi- or fully automated. The paper introduces an approach for the automated verification of parts of buildings, by comparing the design of a building (as-planned model), derived from a Building Information Model (BIM) in an Industry Foundation Classes (IFC) exchange format to a terrestrial laser scanning (TLS) point cloud (as-built model). The approach proposed has three main steps. The process begins with the acquisition of information from the as-planned model in the IFC exchange format; the second step is the automated (wall) plane segmentation from the point cloud. In the last step, the two models mentioned are compared to determine the deviations from the design, and the as-built wall flatness quantification is also executed. The potential of the proposed algorithm is shown in a case-study.

Address

[1] Dept. of Surveying, Faculty of Civil Engineering, Slovak University of Technology in Bratislava, Bratislava, Slovakia

[2] Dept. of Surveying, Faculty of Civil Engineering, Slovak University of Technology in Bratislava, Bratislava, Slovakia

* ***Corresponding author:*** richard.honti@stuba.sk

Key words

- Verification of buildings,
- Point clouds,
- BIM,
- automated verification,
- IFC.

1 INTRODUCTION

One of the most effective techniques for 3D data acquisition of a built environment with a high resolution is terrestrial laser scanning (TLS). With TLS it is possible to perform contactless documentation of the object measured with all of its structural elements without the need to define the characteristic points on the surface of the scanned object. Compared to other surveying methods, TLS is superior due to the speed of its data acquisition, which, with current laser scanners, reaches up to 2 million points per second. The point clouds that result from laser scanning are becoming the initial digital representation of real-world objects and represent a huge amount of data. The accuracy of the determination of the 3D coordinates of single-measured points is several millimeters (depending on the instrument used). The accuracy of the results can be significantly increased using suitable data processing, e.g., regression models, where the modeled points are determined as characteristic points of the regression entities (Vosselman and Maas, 2010). Point clouds are often used as the basic data for the geometry generation for Building Information Models (BIM).

BIM is an intelligent 3D model-based process that enables a builder to more efficiently plan, design, construct, and manage structures and buildings. BIM is an object-oriented 3D model of a building that includes information about the structure, its elements, and their parameters. It includes the development of a virtual data model, which describes not only the architecture but also simulates the construction and operation of the new or renovated object. The aim of BIM is not only the creation of a building model but also the creation of correct, responsible, modifiable, and complete information about the building, including access to this information for all the stakeholders. BIM is based on interoperability and the parametric behavior of its objects, i.e., a change of one parameter generates an aggregated set of changes in all the parameters of the model object (Eastman, et al., 2008; Smith and Tardif, 2009). In terms of data, a BIM model can be split into three main components. It is composed of graphic data, non-graphic data (information), and documents relating to the building project. In addition to the three main components, metadata is recognized as a part of the model. There are different file formats for BIM models since there are many ways in which the data can be

© 2020 The Author(s). This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.Org/licenses/by-nc-nd/3.0/).

handled. The Industry Foundation Classes (IFC) is the most common non-proprietary file format; it aims at providing an open and neutral data format for storing and exchanging BIM data between heterogeneous software applications (ISO 16739-1, 2018).

The measurement of the as-built geometry and its comparison with the design geometry (project plan) is an important part of the construction works. Visual observations and traditional geometric measurements can be time-consuming, expensive, and ineffective. Automatization of the processes of monitoring the progress of construction can contribute to increasing the efficiency and precision of this process. Laser scanners show the potential for supporting the automatic tracking of the progress of construction.

This paper aims to demonstrate the potential use of the BIM model and TLS point clouds for the automated verification of building components. The introduction of the paper gives a short description of the technology of TLS and briefly introduces the concept of BIM models. The acquisition of information from BIM models in the IFC exchange format using a novel algorithm is depicted in the second section of the paper. The next section describes the segmentation process of a wall's plane segments from a point cloud. Then, the deviations between the designed geometry and the as-built geometry are calculated, with the quantification of the flatness of the as-built model's wall. For the automation of this process, a standalone application based on Matlab® software is proposed. The described methodology is presented in a case study in the last section of the paper.

2 SCAN-VERSUS-BIM

In the design and construction phases of structures, the use of BIM models has dramatically increased, but they are not used very often in the operation and maintenance phases. This can be the result of various factors, e.g., the accuracy and the level of detail gathered from a structure's models do not always represent the as-built conditions needed by the maintenance personnel. Furthermore, numerous modifications can occur after the construction that are rarely documented. For the above-mentioned and other reasons, there is a need in some cases for the digitalization to a high degree of detail of buildings after their construction. TLS represents a relatively fast and simple solution to this problem. With the technology of TLS, the as-built conditions of BIM models and changes from a design can be documented and verified (Giel and Issa, 2011).

One of the main applications of TLS today is the reconstruction of as-built 3D BIM models from scanned 3D point clouds, the so-called Scan-to-BIM process. The other process is the Scan-versus-BIM (as an analogy to Scan-to-BIM), which compares the scanned point clouds of built structures with the designed BIM model. A few research teams have already demonstrated the potential value of Scan-versus-BIM processes for tracking the progress and quality control of the dimensions of structural works (Bosché, et al., 2015).

Generally, the first step of the Scan-versus-BIM procedure is to align the as-built model (scanned point clouds) into the same coordinate system as the as-planned model (BIM model). The alignment can be done in several ways, e.g., it can be based on overlaps of the geometric shapes found in each model, the use of some survey points, or the use of other semi- or fully automated registration techniques (Kim, et al., 2013). The second step is the calculation of the deviations. Specifically, the minimum distance from each point of the scanned point cloud to its nearest surface in the BIM is calculated. Other methods compute the distance along user-specified directions, e.g., the X, Y, Z direction of the common coordinate system or the direction of a surface normal. Occasionally, point cloud segmentation is performed to divide the point cloud into objects recognized in the as-planned BIM model, i.e., the points of the as-planned model are matched with the 3D model object, most often with the closest object based on some criterion (e.g., the proximity and normal similarity of a surface). The last step is the visualization of the calculated deviations with color maps, counter lines, etc. (Bosché and Guenet, 2014).

The current trend is the automatization of the steps of the Scan-versus-BIM process. One of the options for the automated geometric inspection of structures is to compare the selected geometric shapes (e.g., walls – plane segments) from the designed BIM model with the as-built geometry gathered from TLS point clouds.

3 GEOMETRY OF A WALL FROM AN IFC EXCHANGE FILE FORMAT

The Industry Foundation Class (IFC) is a platform-neutral, open file format specification that is not controlled by a single vendor or group of vendors. It is an object-based file format with a data model, which is intended to describe building and construction industry data and exchange these data between architecture, engineering, and construction (AEC) software applications. The IFC model specification is open and available. It is registered by ISO and is an official international standard (*ISO 16739:2018 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*) (ISO 16739-1, 2018).

The architecture of the IFC data structure contains four definition layers to which schemas are strictly assigned. The main layer is the domain-specific layer, which organizes definitions according to the industry discipline (building control domain, structural elements domain, electrical domain, etc.). Consequently, it is hierarchically arranged by the shared element layer, which provides more specialized objects and relationships shared by multiple domains. Then the core layer hierarchically follows: it provides the basic structure, fundamental relationships, and common concepts for all further specializations. The fourth layer is the resource definition layer, which consists of supporting data structures (ISO 16739-1, 2018). From a surveyor's point of view, this layer is the most important since it contains the coordinates of the objects within the model as well as other geometric parameters.

In this paper, the Scan-versus-BIM process is accomplished by comparing the wall plane segments between two models. First, the wall object's geometry is acquired from the IFC and defined as *IfcWallStandardCase*. The derivation of the information is executed with a novel algorithm formulated in Matlab® software, which is capable of the automatic identification and extraction of the necessary parameters from the IFC (Fig. 1).

The required data from the IFC for estimating the wall plane parameters are as follows: the reference (initial) point (*IfcCartesian-*

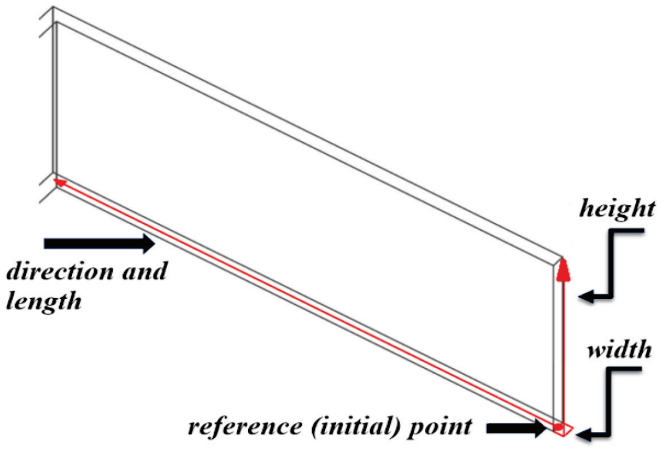


Fig. 1 *Interpretation of the wall parameters in the IFC model*

```
#134= IFCCARTESIANPOINT((52414.1410933094,25630.2336971545,0.));
#136= IFCDIRECTION((-0.537758036394573,-0.843099219720344,0.));
#138= IFCAXIS2PLACEMENT3D(#134,#19,#136);
#139= IFCLOCALPLACEMENT(#125,#138);
#141= IFCCARTESIANPOINT((3825.,-0.));
#143= IFCPOLYLINE((#9,#141));
#145= IFCSHAPEREPRESENTATION(#96,'Axis','Curve2D',(#143));
#148= IFCCARTESIANPOINT((1912.49999999999,1.39266376208980E-12));
#150= IFCAXIS2PLACEMENT2D(#148,#25);
#151= IFCRECTANGLEPROFILEDEF(.AREA.,$,#150,3824.99999999999,139.999999999999);
#152= IFCAXIS2PLACEMENT3D(#6,$,$);
#153= IFCEXTRUDEDAREASOLID(#151,#152,#19,2500.);
#154= IFCCOLOURRGB($,0.709803921568627,0.709803921568627,0.709803921568627);
#155= IFCSURFACESTYLERENDERING(#154,0.,$,$,$,$,IFCNORMALISEDRATIOMEASURE(0.5),IFCSPECULAREXI
#156= IFCSURFACESTYLE('Betonov\X2\00E9\X0\ Tv\X2\00E1\X0\rnice',.BOTH.,(#155));
#158= IFCPRESENTATIONSTYLEASSIGNMENT((#156));
#160= IFCSTYLEDITEM(#153,(#158),$);
#163= IFCSHAPEREPRESENTATION(#98,'Body','SweptSolid',(#153));
#166= IFCPRODUCTDEFINITIONSHAPE($,$,(#145,#163));
#170= IFCWALLSTANDARDCASE('1dzFimrZbEyw6Xo3xbqS2l',#41,'Basic Wall:Obecn\X2\00E9\X0\, 140mm
```

Fig. 2 *Wall object in the IFC exchange format*

Point); the direction (*IfcDirection*) defined by the unit normal vector; the length (*IfcRectangleProfileDef*); the height (*IfcExtrudeAreaSolid*); and the width of the wall (*IfcWallStandardCase*) (Fig. 2).

The parameters needed for the estimating the plane are calculated from the extracted wall geometry. First, the 3D coordinates of the four boundary points and the center point of the wall-side (plane) are calculated using the spatial polar method. Based on these calculated wall points, the plane coefficients are estimated.

The equation for the plane can be described as follows:

$$a.X + b.Y + c.Z + d = 0, \tag{1}$$

where a, b and c are the parameters of the normal vector of the plane; X, Y, Z are the coordinates of a point lying on the plane; and d is the scalar product of the normal vector of the plane and the position vector of any point of the plane. To calculate the elements of the normal vector, the cross product from the boundary points is used.

4 PLANE SEGMENTATION FROM POINT CLOUDS

4.1 Plane segmentation using the modified RANSAC algorithm

After the wall plane segments are extracted from the as-planned IFC model, the planes that belong are segmented from the as-built point cloud data. The segmentation starts with the seed point selection, which is the nearest point from the point cloud to the center point of the extracted plane from the IFC. This seed point is used as a starting point for the plane segmentation. The point cloud segmentation is carried out with a modified version of the algorithm proposed in (Honti, et al., 2018). The modification consists in adding a normal vector-based filter (described in more detail in section 4.2).

The algorithm in (Honti, et al., 2018) is a combination of the elements of the modified RANSAC algorithm and the region-based segmentation method. RANdom SAmple Consensus (RANSAC) is an iterative method for estimating a mathematical model from a data set, which can contain outliers. The algorithm was first published in (Fischler and Bolles, 1981). The idea of RANSAC is to identify the outliers in the data set and estimate the parameters of a mathematical model using only the inlier points (i.e., the points excluding the outliers). Region-based segmentation methods use neighborhood information to merge close points into regions that have identical properties with some tolerance.

In the mentioned algorithm (Honti, et al., 2018), the first plane is estimated from the 10 nearest points to the selected seed point. For estimating the plane, orthogonal regression, which minimizes the perpendicular distances to the estimated plane, is used. The solution is based on the general equation of a plane (Eq. (2)). To calculate the elements of the normal vector, the Singular Value Decomposition (SVD) is used:

$$, \quad \boldsymbol{A} = \boldsymbol{U}.\boldsymbol{\Sigma}.\boldsymbol{V}^T, \tag{2}$$

where $\boldsymbol{A}$ is the design matrix with dimensions of nx3; n is the number of points used for the calculations. The column vectors of $\boldsymbol{U}$ are normalized eigenvectors of the matrix . The matrix contains eigenvalues on the diagonals. The normal vector of the regression plane is the column vector of $\boldsymbol{V}$ corresponding to the smallest eigenvalue from . The plane is then defined by the elements of the normal vector a, b and c. Parameter d is calculated by fitting the elements of the normal vector and the coordinates of the centroid of the point cloud according to the formula

$$d = -(a.X_0 + b.Y_0 + c.Z_0). \tag{3}$$

The orthogonal distances of the points from the plane are calculated as follows:

$$\boldsymbol{d}_p = \frac{|(a.X_0 + b.Y_0 + c.Z_0 + d)|}{\sqrt{a^2+b^2+c^2}}. \tag{4}$$

The standard deviation is calculated as characteristics of the plane using the orthogonal distances of the inlier points from the plane by the formula (5):

$$\sigma_\rho = \sqrt{\frac{\boldsymbol{d}_{p,\rho}^T.\boldsymbol{d}_{p,\rho}^T}{n-1}}, \tag{5}$$

In the next step, the estimated regression model is tested against the nearest neighbors, while the points lying on this plane (inliers) are identified. The plane re-estimation is performed using all the inliers, and it is done iteratively with a gradual increase in the neighboring points tested. The re-estimation of the plane is repeated until the size of the plane stops increasing, so there are no more inliers for the selected plane, or there are no more points for testing in the point cloud.

The segmentation procedure described is performed k times, where k represents the number of planes to be segmented from the point cloud. In this case, it is the number of verified planes in the IFC model.

The process of the algorithm is presented in Fig. 3, which shows the scanned point cloud (*top-left*); then the 102,400 nearest points (*top-right*) to the selected seed point are shown, followed by the 1,638,400 nearest points (*bottom-left*), and the inliers for the chosen plane (*bottom-right*) in the red color, so the results of the segmentation are shown.

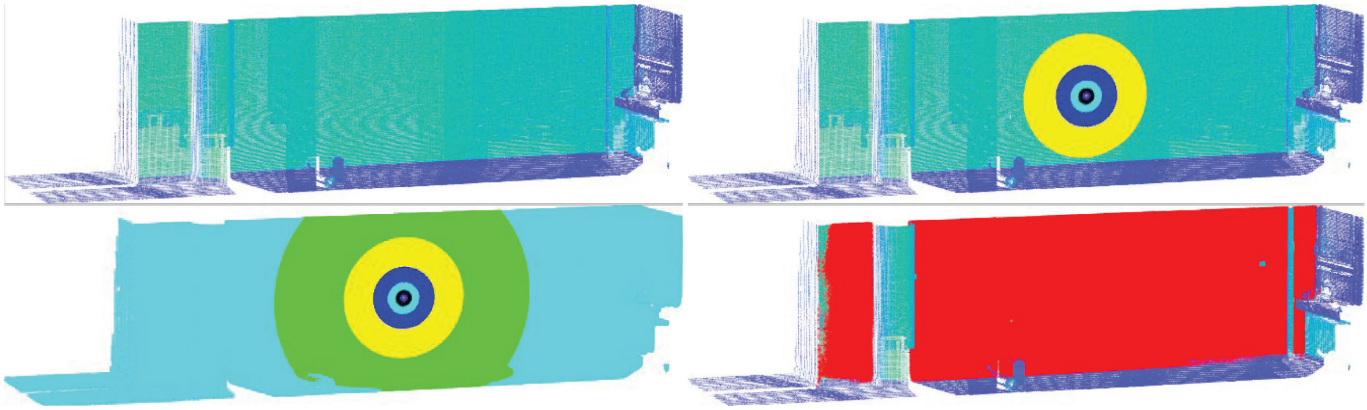


Fig. 3 *The plane segmentation procedure using the depicted algorithm*

4.2 Normal vector-based filtering

In the segmentation algorithm described in the previous section, distance-based filtering is used for the inlier identification, i.e., the inlier points for the estimated planes are the points which are closer to the regression plane than a selected threshold. Since in practical cases, the walls show various deviations from the ideal designed shape, it is necessary to set the threshold number in the distance filter to several centimeters. Therefore, in some cases the points from the surrounding planes or structural elements are also considered as inliers; however, they do not belong to the selected plane (Fig. 4). For that reason, in some cases, it is necessary to add a normal vector-based filter.

The first step after performing the plane segmentation using the algorithm in section 4.1 is to estimate the normal vector at each point of the segmented point cloud for the given plane; small local surfaces (planes) calculated from the *k*-nearest neighbors should be used. For estimating local planes, orthogonal regression is used. Then the angle between the normal vector of the regression plane from the modified RANSAC algorithm (Sec. 4.1) and the normal vector in each point are calculated according to the formula (6)

$$\Delta norm_i = acos(\boldsymbol{n}_{regpl} \cdot \boldsymbol{n}_{p_i}), \tag{6}$$

where is the normal vector of the regression plane; is the normal vector of the local plane at the testing point. The points where this value is less than a selected threshold value are then considered as inliers. This practically means that if a value 5 is chosen as a threshold, only the points where this angle is less than 5 degrees meet the criterion. In this step several outliers are eliminated, e.g., the points of the surrounding planes, the points on the light switch, the skirting board at the bottom, the door frame, and also some points on the door. Fig. 5 shows the results of the segmentation after the normal vector-based filtering is executed.

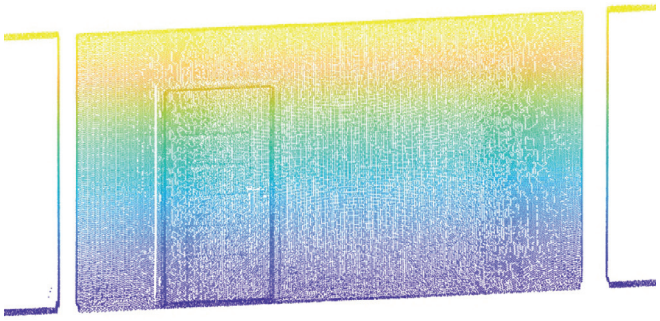


Fig. 4 *Segmented plane before the normal filtering*

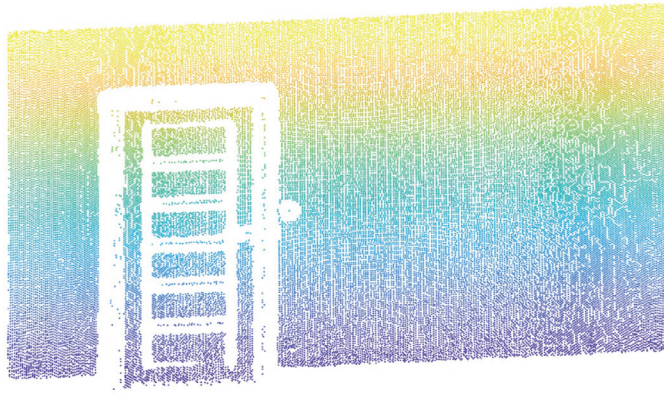


Fig. 5 *Segmented plane after the normal filtering*

5 DIFFERENCE MODEL GENERATION

Fig. 6 shows the process of how the algorithm is developed. The algorithm requires that the 3D point cloud and the BIM model are registered in the same coordinate system.

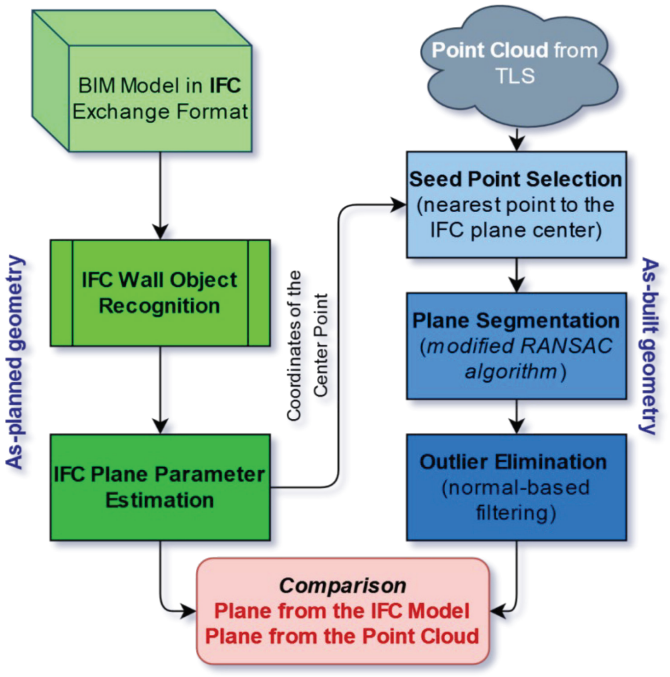


Fig. 6 *Flowchart of the proposed algorithm*

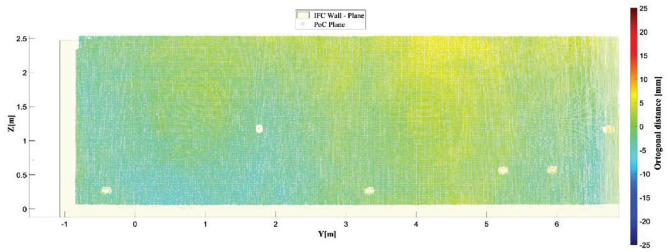


Fig. 7 *Comparison of the IFC plane and the point cloud plane*

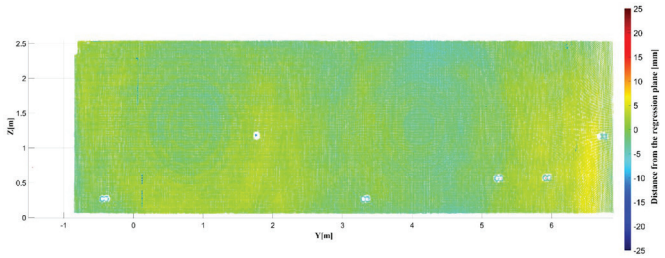


Fig. 8 *Visualization of the quantification of the flatness of the wall*

At this point, the parameters of the as-planned plane are calculated from the information derived from the BIM model, and the plane points that belong are segmented from the scanned point cloud. The next step of the proposed algorithm is a comparison of the IFC plane and the point cloud plane by calculating the orthogonal distances between these two models.

In Fig. 7, the orthogonal deviations are visualized between the IFC plane and the segmented plane from the point cloud. The IFC plane is shown with a bright transparent yellow square, and the segmented points are visualized by color depending on the distance from the IFC plane. The positive values of these deviations are for the points "over" the IFC plane (closer to the scanning position) and the negative values are the points "under" the IFC plane.

In the last step of the algorithm, the quantification of the flatness of the wall is performed (Fig. 8); the best-fit regression plane for the segmented points is estimated using orthogonal regression and calculating the distances for each segmented point to this regression plane.

The whole process (Fig. 6) is repeated semi-automatedly, while all the wall objects from the IFC model are geometrically verified. The results are the spatial deviations between the planes of the design model (IFC) and the as-built geometry (scanned point clouds) in every point of the point cloud, and the quantification of the flatness of the wall.

In addition to these results, the parameters of the estimated planes from both models are compared. The parameters compared are the elements of the plane's normal vector (*a, b, c*) and parameter *d* of the plane. Further, the roll and the offset between the two models are calculated according to formulas (7) and (8):

$$roll = acos(|a_{IFC}.a_{PoC} + b_{IFC}.b_{PoC} + c_{IFC}.c_{PoC}|), \quad (7)$$

$$offset = (|d_{IFC}|-|d_{PoC}|), \quad (8)$$

where and are the parameters of the IFC plane, and and are the parameters of the segmented regression plane from the point cloud according to the general equation of the plane (1).

In the event the building tolerances are provided, a comparison of the deviations between the design and the as-built geometry with these tolerances can be executed to determine if the tolerances have not been exceeded.

5.1 Development of a standalone application

A computational application was developed for the automation of the verification procedure. The application was created as a standalone application, and its graphic user interface was designed in Matlab® software. Matlab runtime is needed for the execution of the application.

The dialog window of the application is divided into 2 main parts (Fig. 9). The first part on the left side serves for loading the input files and adding the input parameters. The input files are the BIM model in the IFC exchange format and the point cloud. The input parameters are the threshold values for the distance and normal filtering. The verification process is launched by pressing the *Run* button.

According to the results, the visualization of the comparison between the two models (Fig. 6) and the quantification of the flatness of the wall (Fig. 7) for each plane are saved into a figure (*.tif) to the selected working directory. The second part of the dialog window is the table on the right side, where the parameters for each plane are shown. The first dialog window table contains the number of the plane, the parameters of the normal vector (*a, b, c,* and *d*) for the IFC plane and the PoC plane, the roll and the offset, the maximal and the minimal deviations, and the average deviation between the two planes. The second table contains the parameters from the quantification of the flatness of the wall, i.e., the maximal, minimal, and average deviations between the points of the segmented plane and the regression plane.

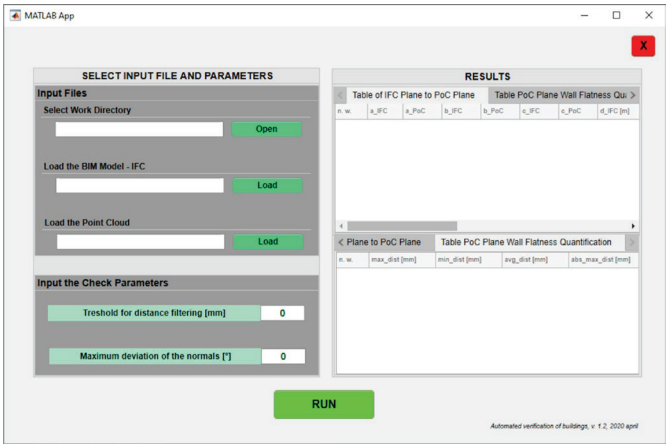


Fig. 9 *Dialog window of the application developed*

6 CASE STUDY

For experimental testing of the above-proposed algorithm, a BIM model in the IFC exchange format and a scanned point cloud of an apartment (Fig. 10) was used. The apartment consists of 5 rooms with 25 walls (the internal plane segment for each wall was considered). The scanning was performed with the Trimble TX5 3D laser scanner. The scanning was executed with a resolution of 10 mm/10 m. The distance between the scanner and the scanned object was always below 5 m. The whole point cloud of the apartment consists of approximately 10 million points. With the mentioned scanner and considering the conditions (maximum distance of the scanned objects from the scanner, resolution, and quality setting of the scanner, etc.) during the measurement, the accuracy of a single measured point was less than 4 mm in all the cases. For practical applications, it is necessary to include the transformation error (transformation of the point cloud into the model's coordinate system) by applying the error propagation law.

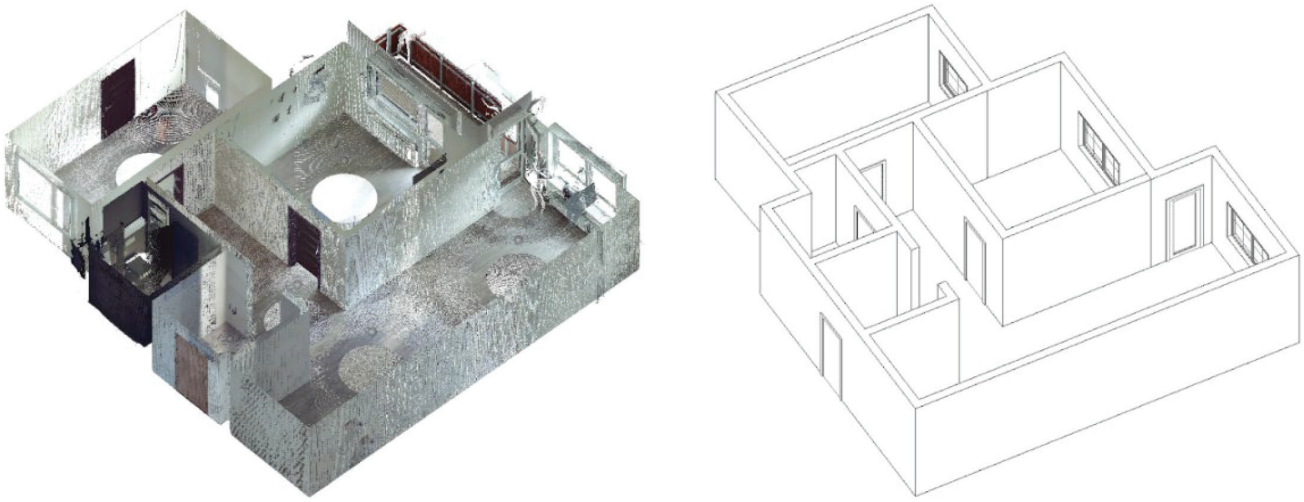


Fig. 10 *The scanned point cloud (left) and the generalized BIM model in the IFC exchange format (right) of the case-study object*

The results from the comparison of the two models for the planes, which show the largest deviations, are shown in Tabs. 1 and 2. Tab. 1 shows the comparison between the plane parameters (*a, b, c,* and *d*), and Tab. 2 shows the roll, the offset, and the maximal, minimal, and average deviations between the two planes.

Tab. 1 *Comparison of the plane parameters*

n	a_{IFC}	a_{PoC}	b_{IFC}	b_{PoC}	c_{IFC}	c_{PoC}	d_{IFC} [m]	d_{PoC} [m]
1	0.537	0.542	0.844	0.840	0.000	-0.003	-7.225	-7.232
2	0.842	-0.841	-0.540	0.541	0.000	0.001	1.497	-1.500
3	-0.846	0.842	0.533	-0.540	0.000	0.001	5.799	-5.797
4	0.536	-0.536	0.844	-0.844	0.000	0.002	-1.282	1.284
5	-0.844	-0.842	0.536	0.539	0.000	0.000	8.622	8.604
6	-0.842	-0.840	0.539	0.543	0.000	-0.001	1.772	1.775
7	-0.539	-0.539	-0.842	-0.842	0.000	0.000	3.184	3.188
8	-0.539	0.535	-0.842	0.845	0.000	0.005	5.160	-5.158
9	-0.842	0.840	0.539	-0.542	0.000	0.000	1.023	-1.001
10	-0.530	-0.537	-0.848	-0.843	0.000	-0.001	6.272	6.352

Tab. 2 *Parameters of the plane deviations*

n	roll [°]	offset [mm]	max_dev [mm]	min_ dev [mm]	avg_ dev [mm]	abs_max_ dev [mm]
1	0.4021	-6.8	36.1	-62.9	-1.3	62.9
2	0.0810	-3.1	33.6	-47.6	-0.3	47.6
3	0.4768	1.6	53.8	-55.1	-16.9	55.1
4	0.0875	-1.9	39.6	-37.4	3.9	39.6
5	0.2407	18.5	45.3	-41.9	0.9	45.3
6	0.2606	-2.8	56.7	-30.9	-6.6	56.7
7	0.0399	-4.5	26.1	-40.0	-1.8	40.0
8	0.4192	2.0	12.6	-74.2	-9.6	74.2
9	0.2052	22.9	77.3	-22.4	2.1	77.3
10	0.5200	-49.7	36.7	-37.5	-11.8	37.5

The maximal roll between the 2 planes is 0.52°; the maximal offset is 49.7 mm in the case of plane 10. The maximal deviation is 74.2 mm in the case of plane 8.

Tab. 3 shows the results from the quantification of the flatness of the wall, where the maximal, minimal, and average distances between the regression plane and the segmented points are shown; they reached from -34.6 mm to 34.9 mm in all the cases of the object tested.

Tab. 3 *Quantification of the flatness of the wall*

n	max_dev [mm]	min_dev [mm]	avg_dev [mm]	abs_max_dev [mm]
1	28.7	-16.0	-0.4	28.7
2	30.5	-29.5	0.1	30.5
3	34.9	-34.6	0.2	34.9
4	17.6	-29.4	-2.0	29.4
5	29.3	-21.2	0.2	29.3
6	23.2	-17.1	0.2	23.2
7	21.8	-24.3	-1.4	24.3
8	18.2	-29.2	2.8	29.2
9	27.1	-27.7	0.3	27.7
10	19.5	-18.7	-6.9	19.5

CONCLUSION

The technology of TLS and the BIM models offers to improve the reliability, efficiency, and completeness of geometric checks of structures. This paper focused on the geometric verification of building components using geometric shapes (planes) and proposed an algorithm that integrates BIM models in the IFC exchange format and scanned point clouds to automate this procedure. First, the walls are recognized in the IFC model; then the parameters of the plane segments belonging to these walls are calculated. In the next step, the planes from the scanned point cloud belonging to the planes in the IFC model are automatically segmented using distance-based and

normal vector-based filtering. Next, the as-planned planes from the IFC and the as-built planes from the point clouds are compared, and the orthogonal deviations are calculated between them. In the last step, the quantification of the flatness of the wall for each wall is performed. The proposed algorithm was implemented in a stand-alone application for its automation. The user can easily execute the geometric verification of the wall plane with the application developed. The experimental testing of the algorithm on a model of an apartment, along with the results, is also described.

Acknowledgments

This work was supported by VEGA project 1/0506/18 "Development of an Algorithm for Automated Quality Check of Construction Work in BIM Environment", and funded by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic.

REFERENCES

Bosché, F. – Guenet, E. (2014) Automating surface flatness control using terrestrial laser scanning and building information models. Automation in Construction, 44, 212-226. DOI: 10.1016/j.autcon.2014.03.028

Bosché, F. – Guillemet, A. – 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 (B), 201-213. DOI: 10.1016/j.autcon.2014.05.014

Eastman, Ch. – Teicholz, P. – Sacks, R. – Liston, K. (2008) BIM Handbook. 1st ed. John Wiley & Sons, NJ, USA.

Fischler M. A. – Bolles R. C. (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of ACM, Vol. 24, No. 6, pp. 381–395.

Giel, B. – Issa R. R. A. (2011) Using laser scanning to access the accuracy of as-built BIM. In: Proceedings of ASCE International Workshop on Congress on Computing in Civil Engineering, 665-672. DOI: 10.1061/41182(416)82

Honti, R. – Erdélyi J. – Kopáčik A. (2018) Plane segmentation from point clouds. Pollack Periodica, 13 (2), 159–171. DOI: 10.1556/606.2018.13.2.16

Kim, Ch. – Son, H. – Kim, Ch. (2013) Fully automated registration of 3D data to a 3D CAD model for project progress monitoring. Automation in Construction, 35, 587-594. DOI: 10.1016/j.autcon.2013.01.005

Smith, D. K. – Tardif, M. (2009) Building Information Modeling, A Strategic Implementation Guide. 1st ed. John Wiley & Sons, NJ, USA.

Vosselman, G. – Maas, H.G. (2010) Airborne and Terrestrial Laser Scanning, Dunbeath: Whittles Publishing, 2010. 318 pp. ISBN 978-1904445-87-6

ISO 16739-1:2018 "Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema".