

## AUTOMATION OF CYLINDER SEGMENTATION FROM POINT CLOUD DATA

<sup>1</sup> Richard HONTI, <sup>2</sup> Ján ERDÉLYI, <sup>3</sup> Alojz KOPÁČIK

<sup>1,2,3</sup> Department of Surveying, Faculty of Civil Engineering  
Slovak University of Technology in Bratislava, Radlinského 11  
810 05 Bratislava, Slovakia

e-mail: <sup>1</sup> richard.honti@stuba.sk, <sup>2</sup> jan.erdelyi@stuba.sk, <sup>3</sup> alojz.kopacik@stuba.sk

Received 19 December 2018; accepted 3 March 2019

**Abstract:** The results of terrestrial laser scanning are point clouds, which are becoming an increasingly common initial digital representation of real-world objects. Since point clouds in the most cases represent a huge amount of data, automation of the processing steps is advisable. The paper brings a short review of the most reliable methods of cylinder extraction. An innovative algorithm is proposed for an automated detection of cylinders and also for estimating their parameters from 3D point cloud data. The method was tested on the complex point clouds of pipelines. The proposed algorithm was implemented to a standalone application based on MATLAB® software.

**Keywords:** Automated data processing, Point cloud, Cylindrical segmentation, Terrestrial laser scanning

### 1. Introduction

The technology of Terrestrial Laser Scanning (TLS) is one of the most effective technologies for the acquisition of 3D information of the measured objects. The main advantage of TLS is that it allows contactless documentation of the measured object with all its structural elements without the need to define some characteristic points on the surface of the scanned object. The spatial position of the points is calculated from the horizontal and vertical angles and the slope distance measured by the instrument. The next advantage of this technology is the speed of the data acquisition (scanning), which reaches with current scanners 2 million points per second, so the time needed for measuring an object is significantly reduced. The result of a measurement by TLS is an

irregular set of measured 3D points lying on the surface of the measured object, i.e. the so-called Point Cloud (PoC). Point clouds (*Fig. 1*) are becoming an increasingly common initial digital representation of real-world objects [1]-[3]. *Fig. 1* shows the PoC colored by real colors, according to the photographs acquired by the internal camera of the instrument.



*Fig. 1.* Point cloud

Point clouds have an important role in creating of high-quality 3D models of objects in a variety of areas, e.g. Building Information Modeling (BIM) [4], interior (exterior) design, urban information systems, documentation, 3D cadaster, etc. This is due to the popularity of affordable and accurate scanning equipment that can quickly digitize the geometry of a real-world object. With the development of hardware, the development of data processing algorithms is also very important, since the resulting point cloud data can often contain millions of 3D points, so manual processing is often a difficult and time consuming task [3], [5].

Most of the objects in industrial environment are composed of basic geometrical shapes, as planes, cylinders, spheres and cones. Often used objects like pipes, sleeves, connectors are shaped like a cylinder. Thus, detection of cylinders in point cloud data can be one of the main processing steps in the 3D model creation. Moreover, cylinder detection could be used in various applications: 3D registration (transformation) of point clouds, pipeline plant modeling, reverse engineering, etc. [6]. Thus, this paper is focused on finding an efficient, robust, and accurate algorithm for the automated cylinder detection in point clouds.

The paper describes the cylinder segmentation process of point clouds and describes some methods and approaches for this purpose. It then introduces the proposed algorithm for the automated cylinder detection and segmentation. The developed algorithm is aimed at extracting cylinders automatically, efficiently and accurately from

scanned data. The paper further presents the testing and the experimental results from the testing of the proposed algorithm and describes the standalone application based on the proposed algorithm. The application was developed using MATLAB<sup>®</sup> software.

## 2. Segmentation of cylinders from point clouds

Since, most of the manufactured objects consist of basic geometrical primitives; one of the most important steps in a point cloud processing is a detection and segmentation of these primitives. The segmentation of PoC is the process of classifying points of the point cloud into the multiple homogeneous regions; the points in the same region will have the same properties [7].

One of the most common geometrical primitives in industrial environment is a *cylinder*. Generally, a cylinder can be described by 3 parameters:

- orientation of the cylinder axis ( $\vec{d}$ );
- a point on the cylinder axis ( $p^o$ );
- radius of the cylinder ( $r$ ).

Many methods investigated the problem of the cylinder detection, segmentation and estimation of their parameters from PoC. Existing methods could be generally categorized into 2 classes: those requiring a pre-segmentation (i.e. removing the outliers from the point cloud) and those, which work directly on point clouds, without the need of the pre-segmentation [6].

The approaches belonging to the first class fit a cylinder to a pre-segmented point cloud and often use an orthogonal regression to minimize the sum of orthogonal distances from the estimated cylindrical surface (e.g. [8]-[10]). These approaches are highly dependent on the accuracy of the pre-segmentation and on the choice of the initial threshold parameters for orthogonal regression. The mentioned reasons make it difficult to use these methods in automated segmentation, they could be rather useful in a manual, respectively in a partially-automated approach, when the segmentation process is performed manually, and consequently the cylinder estimation could be automated.

The second group includes approaches that directly process the measured point cloud without the need of the pre-segmentation. For processing they generally use methods like RANdom SAmple Consensus (RANSAC) (e.g. [11], [12]), Hough transform (e.g. [13], [14]).

RANSAC is an iterative method for estimating a mathematical model from an observed data set, which contains outliers. The algorithm works by identifying the outliers in a data set and estimating the desired model using data that do not contain outliers; therefore, the RANSAC paradigm extracts shapes from the point data and constructs corresponding primitive shapes, based on the notion of minimal sets [5], [15]. According to [11], for cylinders, two data points with their surface normal vectors (estimated by fitting a local plane in  $k$ -nearest neighbors) are enough for fitting a cylindrical model to a point cloud data. However, the user must set several threshold numbers that may vary for different models. Optimal choice of these threshold numbers can be difficult, especially when the point cloud contains noise. The result is dependent

on the random choice of the initial seed point, and in the worst case RANSAC does not converge to a valid solution even when cylinders are present [6].

Hough transform is most commonly used for detecting simple shapes, as lines, circles planes, but could be used also for the cylinder detection [16]. When using the Hough transforms to estimate all the parameters of a cylinder, it results to a 5D space, with high computational and time requirements. For the mentioned reason, the estimation is usually divided into two main steps. First the direction of the cylinder axis is estimated, followed by computing the point on that axis, and the radius of the cylinder.

Another approach is based on the Gaussian sphere. First step is the computation of the normal vector at each point of the point cloud by fitting a local plane in the nearest neighborhood. Then the normal vectors are plotted on the Gaussian sphere. Because all normal vectors on the surface of a cylinder point to the cylinder axis, the Gaussian sphere will show maxima on a big circle. The normal of that circle is the direction of the cylinder axis [17].

Thus, the orientation of the cylinder axis is determined, the other parameters can be subsequently estimated using RANSAC, Hough transform or by algebraic regression, etc.

In the most cases, the individual approaches are combined, e.g. in Chaperon et al. [14], the Gaussian sphere and the RANSAC paradigm is combined for cylinder estimation. The algorithm proposed belongs to the second group; since the cylinder detection is performed directly on the scanned point cloud data (the pre-segmentation is not needed).

### 3. Algorithm for automated cylinder segmentation

Based on knowledge and experiences from the previous research, an algorithm has been developed for the automation of the cylinder segmentation from point clouds.

The algorithm for the automated multiple cylinder segmentation is shown in *Code 1*. The input data for the algorithm is a point cloud, expected radius of the cylinder ( $r$ ); threshold value for radius ( $t_r$ ), distance ( $t_d$ ) and normal ( $t_n$ ) filtering; and the number of cylinders ( $ncyl$ ).

The first step is the computation of the normal vector at each point of point cloud using small local planes calculated from the 3D coordinates of the given point and the  $k$ -nearest neighbors. The orthogonal regression is used for plane estimation. Subsequently, a random initial seed point is selected.

The first cylinder is fitted in 15 nearest neighbors to the selected seed point. The cylinder estimation can be divided into 3 main steps [6]:

- The first step is to compute the orientation of the axis of the cylinder  $\vec{o}$  that is the vector being perpendicular to the normal vectors of these 15 points of the point cloud. To calculate the axis orientation vector, the Singular Value Decomposition is used.
- The second step is the projection of these 15 points on the plane perpendicular to the cylinder axis  $\vec{o}$ . In this plane, these points are distributed as a circle.

- The third step is the parameter estimation of the projected circle (coordinates of the center of the circle and the radius of the circle). For this purpose, algebraic fitting is used, where the algebraic distances are minimized [18]. The coordinates of the center of the circle are transformed back to 3D coordinates, and they are considered as the point  $p^o$  on the cylinder axis. The radius of the circle is also the radius of the cylinder.

Code 1. Algorithm for the automated cylinder segmentation

```

➤ Input: point cloud; cylinder radius  $r$ , threshold  $t_r$  for radius criterion;  $t_d$  for distance criterion,  $t_n$  for normal orientation criterion; number of cylinders  $ncyl$ .

it = 25; - iteration number
for ii = 1 : ncyl
| 1. normal vector computation at each point of the point cloud
| 2. random selection of an initial seed point for cylinder fitting
| index_it = 0;
|   || 3. for index_it = 1 : it
|       || 4. Cylinder fitting to k-nearest neighbors (k=15).
|       ||   -> Find the cylinder axis orientation  $\vec{d}$  by SVD of the matrix of normal vectors.
|       ||   -> 2D projection of these points on the plane with normal  $\vec{d}$ .
|       ||   -> Circle fitting – determining the center ( $p^o$ ) and the radius ( $r$ ) of the circle.
|       || 5. Iterative updating of the inlier points based on normal and distance filtering.
|       || 6. index_it = index_it + 1.
|   ||end
|   || 7. if index_it == it
|       || 8. Validation phase – convergence computing & radius filtering.
|       || 9. if konverg <  $\epsilon$  && num_inl > 50
|           ||   || Validation successful – Detected cylinder is considered as a reliable one
|           ||end
|   ||end
end

➤ Results: set of parameters for the detected cylinders ( $p^o, \vec{d}, r$ ); segmented PoCs for each cylinder.

```

These 3 steps above are applied iteratively for the inliers. Inlier points in individual iteration are updated based on the distance and normal filtering.

Criteria for filtering are formulated as follows:

$$\left( \left| \Delta v z d_i \right| < \frac{r}{t_d} \right) \& \left( \left| \Delta n o r m_i \right| > t_n \right), \quad (1)$$

where  $\Delta v z d_i$  is the orthogonal distance of the testing point to the cylinder surface;  $\Delta n o r m_i$  is the angle between the normal vector of the local plane in the testing point and between the perpendicular vector to the axis of the cylinder detected in the testing point. Parameters  $t_d$  and  $t_n$  are the threshold numbers.

In the distance filter ( $\Delta v z d_i$ ) the criterion is  $r/t_d$ . Practically it means that if a cylinder with radius of 20 cm is considered, and  $t_d$  equals to 50, only the points, which are closer than 4 mm to the cylinder detected are considered as inliers for the cylinder fitting. For the normal vector orientation filter ( $\Delta n o r m_i$ ), the criterion is  $t_n$ . If value 4.5 is chosen for it, only the point where the angle between 2 vectors is less than 4.5 degree meets the criterion. Small values for the threshold numbers make the process converge more slowly, while with large values we get only small number of inliers, because the criterion is too strict. Therefore, values  $t_d=50$  and  $t_n=4.5$  can be considered as optimal, which were determined empirically.

Based on the mentioned criteria the inliers are automatically updated in individual iteration, and the outliers are filtered from the estimation process. After individual iteration the cylinder is recalculated using all the points that meet the specified criterion.

After completing the iterative re-estimation of the cylinder, radius-based filtering is performed. Only the cylinders with radius of  $r \pm t_r$  are considered as reliable ones. Values of  $r$  and  $t_r$  need to be chosen at the beginning of the algorithm. Moreover, the result of cylinder estimation depends on the initial seed point and the surrounding area, and because of a noise in point cloud or point cloud structure, some detected cylinders may not represent any of the characteristic cylinders of the scanned object. To eliminate these cases, a validation step is proposed. The estimation of the parameters of 'good' cylinders converge after a few iterations, so the validation is performed by determining the difference in the cylinder parameters estimated in the last two consecutive iterations (2):

$$|param_i - param_{i-1}| < \varepsilon, \quad (2)$$

where  $param$  are the estimated cylinder parameters;  $\varepsilon$  is the convergence parameter ( $\varepsilon=0.001$ ).

If the estimated parameters meet the previous convergence criterion (2), the cylinder is considered as a reliable one, and the point cloud segmentation process is performed. In case that the point cloud contains another cylinder, the next segmentation cycle is performed.

The major advantage of the proposed algorithm consists of choosing the inlier points iteratively based on 2 criteria (the distance and the normal filter), so the outliers and the noise are sequentially removed from the estimation process. Moreover, 'badly' estimated cylinders are eliminated in the validation step.

One of the main advantages is that the user can perform a cylinder segmentation based on the selected cylinder radius on the beginning of the algorithm. In case of large point clouds, this can mean a significant reduction of the time required for the estimation.

The next advantage is that after an estimated cylinder is considered as a reliable one, the points that belong to the given cylinder are removed from the original point cloud; this means that while searching for the other cylinders, the estimation is executed only on the remaining points of the point cloud. This step significantly reduces the time required for the calculations. In addition, the inlier selection process in individual iteration is performed on the whole point cloud at once (i.e. multiple points are tested if they lie in the estimated cylinder); this speeds up the overall process. Comparing with other approaches, e.g. RANSAC or region growing method, just one point is merged at individual iteration.

3.1. Testing of the proposed algorithm

For the experimental testing of the above-described proposed algorithm, a point cloud of a reference double-cylinder model (Fig. 2) and the point cloud of the part of the pipeline of the swimming pool heating system were used.

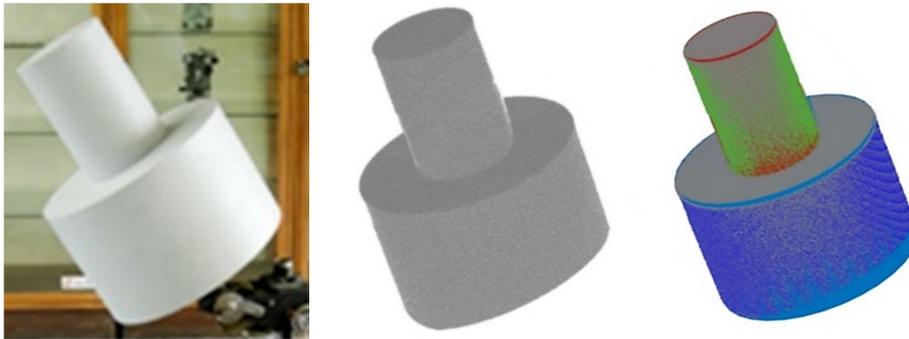


Fig. 2. The reference double-cylinder model (left), the point cloud measured (middle); result of the cylinder segmentation from the proposed algorithm (right)

Fig. 2 demonstrates the point cloud segmentation of the model figure (a double cylinder) using the given algorithm, where different cylinders are differentiated in color (Fig. 2, right).

For the verification of the differences (Table I) among the known parameters (radius) and the estimated parameters using the proposed algorithm were calculated.

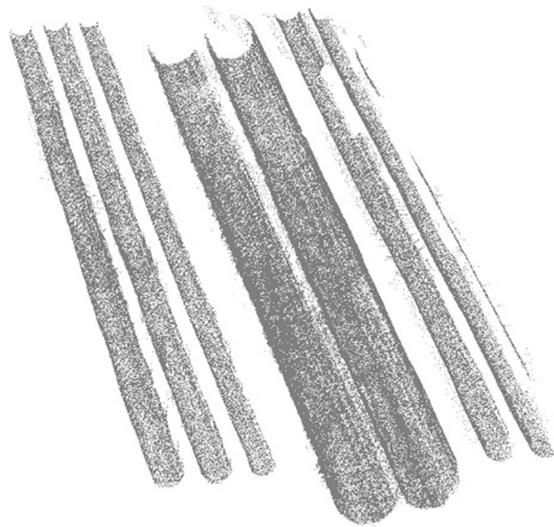
Table I

Comparison of the cylinder parameters

	Reference Values	Values from the Processing	Difference
	[m]	[m]	[mm]
$r_1$	0.2000	0.2004	0.4
$r_2$	0.0900	0.0906	0.6

The reference double-cylinder model was manufactured with the radius of  $r_1=0.2000$  m and  $r_2=0.0900$  m. Values for the radius obtained by processing of point clouds were  $r_1=0.2004$  m and  $r_2=0.0906$  m ( $r_1$  is the lower cylinder,  $r_2$  is the upper cylinder). The difference between the known and the estimated parameters is 0.4 mm for the lower cylinder and 0.9 mm for the upper cylinder. In these deviations the measurement error, the instrument systematic errors, the effect of the environmental conditions, and the registration (of each PoC to the common coordinate system) error are also included.

Next, the PoC of the pipeline system was used for the experimental testing (*Fig. 3*). For the scanning a Trimble TX5 3D laser scanner was used. With the mentioned scanner and considering the measurement conditions, the accuracy of a single measured point was less than 2.2 mm in any cases. The point cloud contains 7 cylinders. The segmentation process was accomplished in 2 phases, because of different radii of the cylinders.



*Fig. 3.* Point cloud measured

In the first phase 2 cylinders with radius of approx. 0.150 m, and in the second phase 5 cylinders with radius of approx. 0.075 m were segmented.

*Fig. 4* shows the experimental result of the processing of the given PoC with the proposed algorithm. Different cylinders are differentiated in color. The whole point cloud consists of 880,815 points. The standard deviation, calculated based on the orthogonal distances of points from the cylinder estimated, was less than 2 mm in any cases.

### 3.2. Software development

The proposed segmentation process places high demands on the workstation used for data processing, as well as on the operator itself. For the automation of the proposed

segmentation procedure, a computational '*PoC\_Cylinder\_Fitting*' application was developed (Fig. 5). The graphical user interface of the application was designed by using MATLAB® software. The application was created as a stand-alone application; however, MATLAB® runtime is needed for its execution. The dialog window consists of three main sections:

- (i) data loading;
- (ii) table with the resulting parameters of the segmented cylinders;
- (iii) figure showing the original point cloud, and the segmented cylinders.

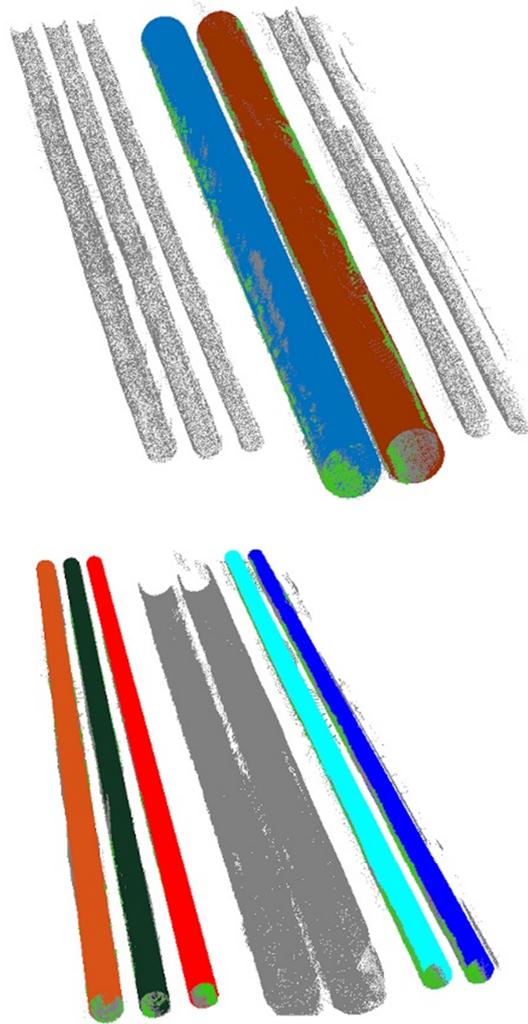


Fig. 4. Point cloud with the segmented cylinders using the proposed algorithm

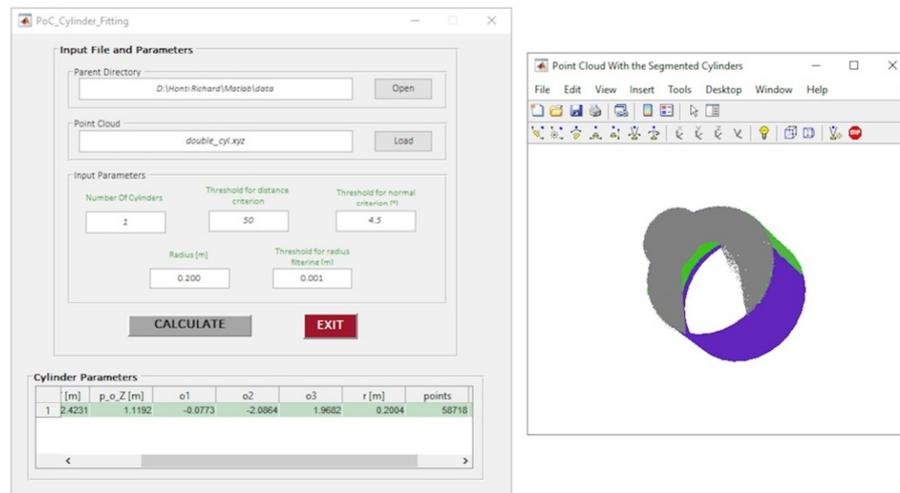


Fig. 5. Dialog window of the 'PoC\_Cylinder\_Fitting' application

In the first section (i), the first step is the parent directory selection, where the input PoC is saved, and the resulted segmented PoCs will be exported. The next step is loading the point cloud file in the \*.txt or \*.xyz file format that contains the spatial coordinates of the points scanned. Before starting the segmentation process, it is necessary to select the input parameters:

- number of cylinders;
- threshold numbers for a distance ( $t_d$ ) and for a normal criterion ( $t_n$ );
- radius ( $r$ ) of the cylinder, and the threshold number for radius filtering ( $t_r$ ).

According to the results of the segmentation process, the individual point clouds for the estimated cylinders are saved into a \*.txt file in a new folder 'Results' in the parent directory, and the cylinder parameters are shown in the 'Cylinder Parameters' table on the bottom side of the application's dialog window (Fig. 5, left-bottom). The table contains the IDs of the cylinders, their estimated parameters ( $p^o$ ,  $\vec{o}$ ,  $r$ ), the number of compatible points of the cylinders, and the standard deviation of the cylinders estimated. A figure (Fig. 5, right), which shows the original point cloud and the separate point clouds for the estimated cylinders, is created for a consistent monitoring of the application process, so the user can visually check whether the individual cylinders are correctly segmented.

#### 4. Conclusion

The paper briefly introduces the possible methods and approaches for the cylinder segmentation from point cloud data. An algorithm for the automated cylinder segmentation from PoC is proposed. The experimental testing and the verification using a reference cylinder model of the proposed algorithm are also described. The proposed

algorithm was implemented as a stand-alone application. The user can easily execute the cylinder segmentation of point clouds with the created application. The results of the application are the segmented points saved into the text files for every cylinder and the parameters of the segmented cylinders are shown in the table of the application's dialog window. The robustness and the accuracy of the results suggest that the algorithm can be used in several applications.

The future work will be to apply the proposed algorithm on large point clouds from several areas, e.g. industrial plants, urban area. Furthermore, this procedure could be extended to other types of geometric shapes, as spheres or cones. The implementation of the algorithm in C++ language is also planned.

### Acknowledgements

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

### References

- [1] Berényi A. Laser scanning in engineering survey, An application study, *Pollack Periodica*, Vol. 5, No. 2, 2010, pp. 39–48.
- [2] Erdélyi J., Kopáčík A., Kyrinovič P., Lipták I. Automation of point cloud processing to increase the deformation monitoring accuracy, *Applied Geomatics*, Vol. 9, No. 2, 2017, pp. 105–113.
- [3] Vieira M., Shimada, K. Surface extraction from point-sampled data through region growing, *International Journal of CAD/CAM*, Vol. 5, No. 1, 2005, pp. 19–27.
- [4] Tkáč M., Mesároš P., Mandičák T. Terrestrial laser scanning, effective technology for creating building information models, *Pollack Periodica*, Vol. 13, No. 3, 2018, pp. 61–72.
- [5] Honti R., Erdélyi J., Kopáčík A. Plane segmentation from point clouds, *Pollack Periodica*, Vol. 13, No. 2, 2018, pp. 159–171.
- [6] Tran T. T., Cao V. T., Laurendeau D. Extraction of cylinders and estimation of their parameters from point clouds, *Computers & Graphics*, Vol. 46, 2015, pp. 345–357.
- [7] Vosselman G., Maas H. G. (Eds) *Airborn and terrestrial laser scanning*, Dunbeath, Whittles Publishing, 2010.
- [8] Benko P., Kós G., Várady T., Andor L., Martin R. Constrained fitting in reverse engineering, *Computer Aided Geometric Design*, Vol. 19, No. 3, 2002, pp. 173–205.
- [9] Taubin G. Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 11, 1991, pp. 1115–1138.
- [10] Lukács G., Martin R., Marshall, D. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation, *European Conference on Computer Vision*, Freiburg, Germany, 2-6 June 1998, In: Burkhardt H., Neumann B. (eds) *Computer Vision Lecture Notes in Computer Science*, Vol. 1406, Springer, pp. 671–686.
- [11] Schnabel R., Wahl R., Klein R. Efficient RANSAC for point-cloud shape detection, *Computer Graphics Forum*, Vol. 26, No. 2, 2007, pp. 214–226.

- [12] Li Y., Wu X., Chrysathou Y., Sharf A., Cohen-Or D., Mitra N. J. GlobFit: consistently fitting primitives by discovering global relations, *ACM Transactions on Graphics*, Vol. 30, No. 4, 2011, Paper No. 52.
- [13] Rabbani T., Van Den Heuvel F., Vosselman G. Segmentation of point clouds using smoothness constraint, Proceedings of the ISPRS Commission V Symposium Vol. 35, Image engineering and vision metrology, Dresden, Germany 25-27 September 2006, pp. 248–253.
- [14] Chaperon T., Goulette F. Extracting cylinders in full 3D data using a random sampling method and the Gaussian image, *Proceedings of the Vision Modeling and Visualization Conference*, Stuttgart, Germany, 21-23 November 2001, pp. 35–42.
- [15] Fischler M. A., Bolles R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of ACM*, Vol. 24, No. 6, 1981, pp. 381–395.
- [16] Liu Y. J., Zhang J. B., Hou J. C., Ren J. C., Tang W. Q. Cylinder detection in large-scale point cloud of pipeline plant, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19, No. 10, 2013, pp. 1700–1707.
- [17] Vosselman G., Gorte B. G. H., Sithole G., Rabbani T. Recognising structure in laser scanner point clouds, In M. Thies, B. Koch, H. Spiecker, & H. Weinacher (Eds), *Proceedings of the ISPRS working group VIII/2: laser scanning for forest and landscape assessment*, Freiburg, Germany, 3-6 October 2004, pp. 33–38. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, working group VIII/2, laser scanning for forest and landscape assessment, Vol. 36, No. 8, Freiburg, Germany, 3-6 October 2004, pp. 33–38.
- [18] Pratt V. Direct least-squares fitting of algebraic surfaces, *Newsletter on ACM SIGGRAPH Computer Graphics*, Vol. 21, No. 4, 1987, pp. 145–152.