Towards cylindrical primitive extraction from industrial laser-scan data by local approximation with refinement

S. Slyadnev, V. Turlapov
sergey.slyadnev@gmail.com, vadim.turlapov@itmm.unn.ru
Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia

We present a novel method for semi-automatic reconstruction of cylindrical primitives from three-dimensional dense point clouds corresponding to the results of laser-scan acquisition. Our method is a part of the general reverse engineering procedure aimed at reconstructing a complete 3D model of an industrial site, such as a petrochemical factory. We use differential properties of a point cloud to restore cross-sections of a cylindrical primitive representing a straight pipe segment. Our method requires user interaction for the selection of cross-sectional circles of the candidate cylinders. In order to make reconstruction more accurate, we refine the extracted radius and axis by taking into account the extended neighborhood of the sample point.

Keywords: *As-built reconstruction, reverse engineering, cylinder fitting, industrial reconstruction.*

1. Introduction

Digital models of industrial facilities, such as refineries, are important in many scenarios, including training, maintenance, renovation of equipment, simulation of different operational situations, data exchange, etc. Existing industrial facilities can be accurately measured with the use of terrestrial laser scanners. Laser scanning systems produce large-scale point clouds with a point density large enough to cover even small details of a measured facility. Typical point clouds representing industrial installations may contain hundreds of millions of points. Automated and accurate transformation of such point clouds to compact 3D representations remains an open problem which is known as *reverse engineering*. A class of reverse engineering problems dealing with large amounts of discrete data representing existing industrial facilities is called *as-built reconstruction*. Oil platforms, factories, buildings are just some of the examples of the facilities that can be reengineered in this field.

There is a growing demand for as-built reconstruction of industrial sites using laser-scan data. As-built reconstruction is aimed to produce a compact three-dimensional model of an operational facility which can be subsequently used in different design and management software (e.g. AVEVA PDMS). Our paper is focused on the reconstruction of cylindrical pipelines. As mentioned by [Rabbani 2006] and [Qiu et al. 2014], a cylinder is one the most frequently used primitives in industrial engineering. This is especially true for processing industries such as petrochemical plants, refineries, nuclear power plants, etc. As a result, automatic and robust methods for detecting and fitting cylindrical primitives are essential for the reconstruction of such sites.

We present a reconstruction procedure which employs differential characteristics of a point cloud as proposed by [Son and Kim 2016]. We demonstrate that the use of differential heuristics alone may introduce a variety of subtle phenomena which complicate the reconstruction process. These problems can be solved by refining the extracted differential characteristics taking into account global cues. Our work is focused on the reconstruction of straight pipe segments. We do not examine pipe connection features such as elbows or tee pipes. Reconstruction of pipe connection features and other primitive types is left for future research.

2. Related work

[Son et al. 2015] conducted a survey on as-built reconstruction methods for industrial facilities, such as petrochemical plants. Their study covers both automatic and interactive reconstruction techniques including approaches adopted in commercial systems. In particular, the authors mention that the *majority of existing commercial tools are interactive in nature.* Although, according to [Benko et al. 2001], in respect of common engineering objects (i.e. those bounded by simple analytic surfaces) significant research efforts are directed at automatic reconstruction with little or no user interaction.

Reverse engineering typically starts with segmentation of an input point cloud into meaningful regions. In general, the segmentation problem is ill-posed, as there is no any objective criterion to measure the segmentation quality. A particular segmentation technique is chosen in function of the target application domain. [Le and Duan 2017] proposed a segmentation framework for the reconstruction of mechanical CAD parts. The authors state that their approach is more robust than the widely adopted random sample consensus (RANSAC) method introduced by [Schnabel et al. 2007]. They identify the model's major directions and perform dimensional reduction before the extraction of various types of geometric primitives. The method is based on the observation that many mechanical models have only a moderate number of primary orientations around which the CAD features reside. These orientations are extracted from Gaussian maps corresponding to plane normals. As-built reconstruction methods actively use Gaussian maps for the recognition of principal axes of an object.

For the automatic decomposition of large point clouds into connected regions, [Rabbani et al. 2006] used segmentation by smoothness constraint which is based on the pre-computed normal field. Segmentation comprises two steps: normal estimation and region growing. This simple technique has found broad application in industrial reconstruction. Region growing is one of the initial stages of our method.

[Benko et al. 2001] utilized the so-called direct segmentation which is an efficient, non-iterative approach. Their segmentation technique is based on the selection of "stable" points within a candidate region. If local characteristics of points within a selected area are unstable, the algorithm deduces that the neighborhood lies between two or more regions.

A popular technique for 3D reconstruction from a point cloud is mesh generation (see [Hoppe et al. 1992]). However, mesh alone is rarely enough, as it lacks information on semantics (CAD features) and does not offer such compact representation as planar, cylindrical, conical, or toroidal primitives. There are hybrid approaches which employ mesh generation at the pre-processing stage for accurate reconstruction. [Masuda and Tanaka Ichiro 2010] presented a system which is based on the conversion of input point clouds into mesh models. After mesh generation the user may perform interactive picking of a seed region which is automatically expanded by the system to one of the conventional primitive types.

[Qiu et al. 2014] presented a method for the reconstruction of pipelines and their joints from industrial point clouds. According to the authors, RANSAC method is not reliable enough to drive the automated as-built reconstruction process. The same applies to the GlobFit method described by [Li et al. 2011], as, according to [Qiu et al. 2014], the GlobFit method may suffer from unreliable initial primitive detection.

[Liu et al. 2013] classify all pipelines from laser scan data as those perpendicular to the ground and those parallel to the ground. They reduce 3D reconstruction problem into a simpler 2D circle detection problem. However, the employed prior knowledge of the ground plane and the assumption about the pipes orientation with respect to this plane may be false.

[Tran et al. 2015] extract cylindrical primitives automatically. They use a dedicated validation stage to accept or reject the fitting results. The authors notice that acquisition noise and point cloud structure may lead to convergence to a wrong solution. Therefore, multiple fitting attempts are necessary to achieve the expected results.

[Son and Kim 2016] presented a fully automatic pipeline reconstruction procedure. Their method employs local approximation with a B-spline surface to derive differential properties of a point cloud. It looks promising; however, the described method may suffer from fluctuations in extracted differential properties due to imperfections of a locally approximating surface. We have reused the ideas presented by [Son and Kim 2016] and enriched them with additional refinement stages aimed to offset the negative effects of local fluctuations.

3. Algorithm

Regardless of whether a software system is interactive or not, there is a common pool of algorithms serving the reconstruction procedure. The "gentleman's set" includes noise smoothing, point cloud filtering, region growing, primitives fitting, feature recognition, and many other utilities. Our intention was to develop a set of auxiliary methods which would help solve a broad range of as-built reconstruction problems. The output of reconstruction should be transferable to any popular CAD system for accurate manual operation. In order to achieve this goal, we have created an interactive system for as-built reconstruction. The system works as a test bench for different reverse engineering techniques, including those which allow for user interaction. In this section, we present the basic implemented approach and highlight several subtle problems that may arise.

3.1 Outline of the algorithm

Our algorithm is outlined as follows:

1. Prepare point cloud:
   a. Filter out near-coincident points in 3D.
   b. Build k-d tree for the initial point cloud.
   c. Build normals for the initial point cloud.
2. Perform coarse segmentation by region growing.
3. Prepare cross sections:
   a. Let the user pick interactively seed points within the area where local characteristics of points are stable.
   b. Approximate each picked point's neighborhood with a B-spline surface.
   c. Build normal plane using differential properties of local approximation surface.
   d. Find points which lie within a certain threshold distance to the normal plane. We call such points a *neighbor band.*
   e. Refine normal plane with respect to the Gauss map of the normal vectors corresponding to the neighbor band points.
   f. Calculate the normal section curve by intersecting the approximation surface with a refined normal plane.
   g. Build an osculating circle for the normal section curve.
   h. Project neighbor band points to the normal plane.
   i. Refine the osculating circle to fit the projected points.
   j. Snap the resulting radius value to one of the predefined radius values of Piping and Instrumentation Diagrams (P&ID) available for the industrial facility.
4. Build a cylindrical surface between the reconstructed circular sections.
5. Trim the cylindrical surface by projecting the points of a region and filtering out the outliers.

For fast nearest neighbor matching we have used the open-source FLANN library presented by [Muja and Lowe 2014]. Normal vectors are calculated by local plane approximation with the use of covariance matrices at each point as described by [Hoppe et al. 1992]. Normal orientation calculated in this way is ambiguous (see Figure 1). However, region growing segmentation and other steps of our method are not sensitive to such ambiguity.

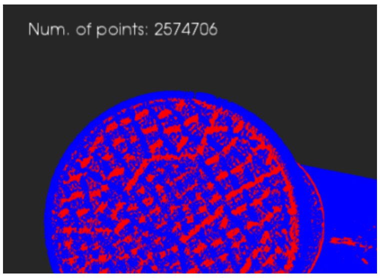


Fig. 1. Ambiguity in the orientation of normal vectors (blue). The original point cloud is shown in red color.

Region growing has been used for decomposition of the initial point cloud into the segments which can be processed individually. Such coarse segmentation is an essential step to enable the work with large industrial point clouds as it allows for piece-by-piece operation, thus relaxing the hardware's requirements. The segmented point cloud is then interactively modified by the user. The user picks a set of seed points to make an assumption on cylindrical primitive's location. The system analyzes local differential characteristics of the selected region and reconstructs a cross-sectional circle which passes through a refinement stage as described in paragraph 3.4. The system is then asked to build an infinite cylindrical surface passing through the set of guess circles. Finally, the infinite surface is trimmed by a subset of tentative points of a region. The trimming is accomplished by checking if the distances from the points to the cylindrical surface are within a threshold value. Figure 2 shows some results obtained with the use of our method on synthetic models.

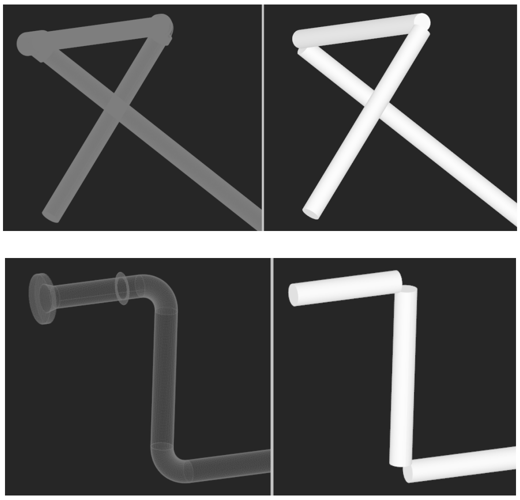


Fig. 2. Some examples of the reconstruction results obtained with the use of our method.

3.2 Data management

The reconstruction process involves several data abstractions which are conveniently represented in an object-oriented programming language such as C++. In addition to the data structures representing input point clouds and resulting primitives, it is convenient to have dedicated data structures for segmentation results, normal vectors, Gauss maps, curves and surfaces employed in reconstruction, etc. In our work, we used the open-source framework [Slyadnev 2017] for data management and visualization. Figure 3 shows the hierarchy of data objects employed in our reconstruction procedure.

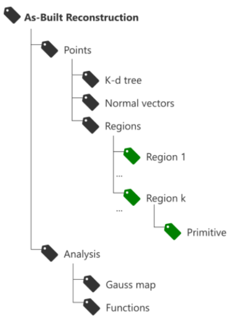


Fig. 3. Hierarchy of the employed data structures. Skeleton objects which persist throughout the reconstruction process are marked with black labels. Objects which are allocated during the reconstruction procedure are marked with green labels.

All inputs and outputs are placed under the *Points* object which represents the initial point cloud. The child objects store the k-d tree for fast neighbors access and the normal vectors calculated at each point. The results of the region growing segmentation are collected under the *Regions* group. A region is nothing but a set of indices pointing to the input cloud elements. Once the reconstruction is done, the resulting primitives can be saved to STEP (ISO 10303) file allowing for interoperability with any CAD system.

3.3 Extract differential properties

Some popular computer vision techniques for primitive fitting are RANSAC and Hough transform. These methods generally avoid using differential characteristics of a point cloud such as principal curvatures. In our method, we calculate the differential properties of an interactively picked area. Having these properties allows for *recognition prior to fitting*. Recognition lets the system “guess” whether the picked region constitutes a cylinder or some other type of primitive. Significant oscillations in the local characteristics let the system guide the user to choose another region which is more “stable”.

Differential properties of a point cloud segment can be estimated by local approximation. Such properties as principal curvatures are very sensitive to fairness of the used approximation surface. In this work, we have adopted *thin plate spline (TPS)* technique based on OpenCascade geometric kernel (see [Slyadnev 2014] for the overview). The used approach is similar to the one described by [Hegland et al. 1997].

3.4 Refinement of cylindrical section

According to [Kim and Son 2016], differential characteristics of a locally approximating surface patch can be used to extract the main properties of a reconstructed cylindrical surface, i.e. its radius and axis. We have found however that using differential properties alone is a troublesome technique. The reconstruction approach which is exclusively based on differential properties of a point cloud is impractically sensitive to the quality of a locally approximating surface. Intuitively, such an approach may not provide reliable results as it uses inherently local (thus unreliable) geometry characteristics to derive global shape properties. Depending on a surface approximation technique, it may happen that the obtained radius and the axis direction are significantly distorted even for a regular point cloud without any noise. Figure 4 illustrates the osculating circle which does not fit in the intuitively expected shape of the cylinder.

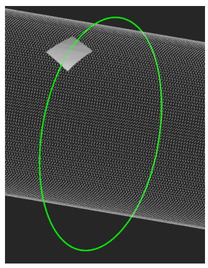


Fig. 4. An example of an inaccurate osculating circle whose quality suffers from oscillations in a local approximation surface. The radius of the osculating circle is greater than expected.

Figure 5 illustrates unexpected inclinations of normal planes. The inaccuracy of a normal plane means inaccuracy of the detected axis direction.

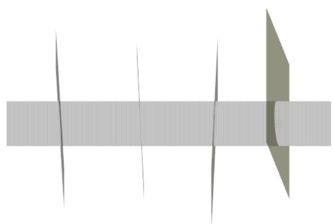


Fig. 5. Unexpected inclinations of normal planes due to small fluctuations of local approximation surfaces.

Both problems are caused by insufficient fairness of a locally approximating surface. It is known from geometric modeling practice that the formalization of such "fairness" is not straightforward. Summarizing the detected phenomena, we conclude that the differential properties of the approximating surface are "weak" heuristics which cannot be directly used in reasoning about the global pipe's properties. Apparently, both the radius and the axis of the cylindrical primitive have to be refined taking into account a "more global" behavior of the point cloud in a sufficiently larger subregion.

3.4.1 Axis refinement

First, we sharpen the orientation of the normal plane. As a global cue for refinement, we use the neighbor points lying near the original section plane (Figure 6). The corresponding normal vectors constitute a Gauss map whose vectors should be predominantly orthogonal to the cylinder axis.

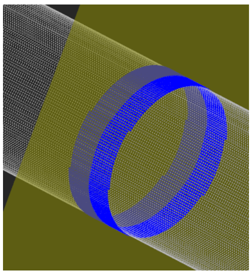


Fig. 6. Neighbor band points as a global cue for refinement.

For the unknown axis coordinates (x_1, x_2, x_3), a residual value r for a single Gauss map element $\mathbf{u} = (u_1, u_2, u_3)$ can be calculated as a dot product $r = x_1 u_1 + x_2 u_2 + x_3 u_3$. The perfect fit is achieved for $\sum_{i=1}^{N} r_i = 0$, where N is the number of elements in the Gauss map (the number of points in the neighbor band). The perfect fit criterion can be formulated for each component individually as $x_k \sum_{i=1}^{N} u_i = 0$ for $k = 1,2,3$. In matrix notation, this can be reformulated as a homogeneous linear system $A\mathbf{x} = 0$, where A is the diagonal matrix. Minimizing $|A\mathbf{x}|^2$ under the constraint $|\mathbf{x}| = 1$ using the Lagrange multipliers yields

$$\mathbf{x}^T A^T A \mathbf{x} + \lambda(\mathbf{x}^T \mathbf{x} - 1) = 0.$$

Derivation by $\mathbf{x}$ yields

$$A^T A \mathbf{x} + \lambda \mathbf{x} = 0.$$

Therefore, we end up with the eigenvalue problem for $A^T A$ matrix. Figure 7 shows an example of axis refinement.

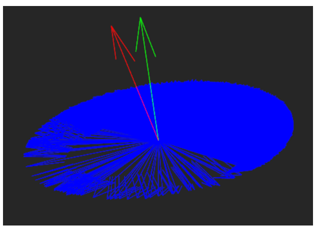


Fig. 7. Axis refinement using the Gauss map of neighbor band points. The original axis direction (red) is orthogonalized (green) in accordance with point cloud normals.

3.4.2 Cross-section refinement

The refined normal plane is used for computation of the intersection curve on the locally approximating surface. Inspired by [Son and Kim 2016], our intention was to use the osculating circle for this curve as a cross-section of the target cylindrical primitive. However, such an approach is not sufficiently reliable due to possible oscillations of the approximating surface leading to a wild shape of the intersection curve (Figure 8).

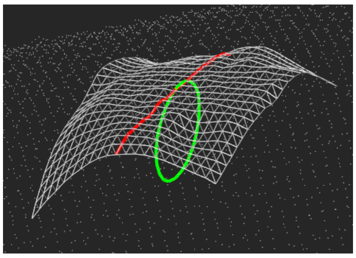


Fig. 8. Oscillating intersection curve (red) due to the insufficient fairness of the approximation surface (white wireframe). The osculating circle is shown in green color.

For circle refinement, we use simple gradient descent method with Armijo rule for adaptive step selection. The radius and the (u, v) center coordinates of the original osculating circle are passed to the optimizer as the initial guess. Other approaches, such as direct circle fitting proposed by [Coope 1993] also look promising. One example of osculating circle refinement is shown in Figure 9.

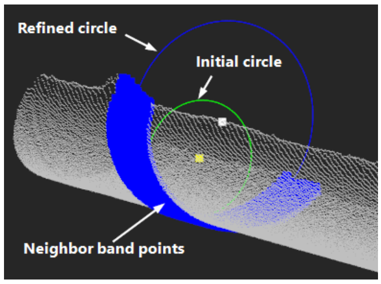


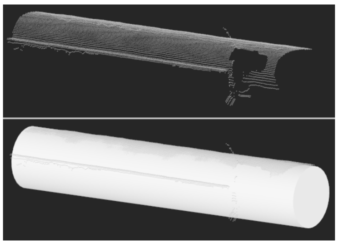
Fig 9. The result of refinement (blue circle) applied to the original osculating circle (green) used as the initial guess in local optimization.

4. Discussion and future research

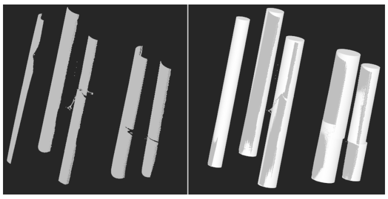
Fully automatic as-built reconstruction is a challenging problem. We believe that the user's intervention cannot be avoided in general case as no algorithm can reliably substitute an experienced human engineer. On the other hand, there is a huge amount of manual operations which consist in searching for stereotypical shapes in the input laser-scan data and selecting the best fitting primitives for the unambiguous regions. Such operations require many man-days of qualified human engineer's work, so their automation is greatly beneficial. We have presented the algorithm for semi-automatic extraction of cylindrical primitives from dense laser-scan data. The algorithm is based on local analysis of a point cloud region with its user-driven completion into a segment of a straight cylindrical pipe.

We have found that differential characteristics are weak cues even if applied to synthetic point clouds without noise. Two such weak characteristics were used in our work: the osculating circle and the axis of the normal plane. Any such differential characteristic requires refinement by global cues. E.g. center and radius of osculating circle were refined by simple gradient descent method using the neighbor band points.

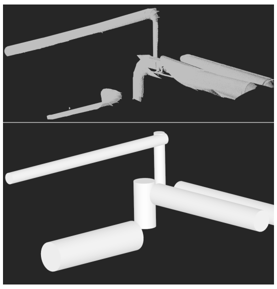
Figure 10 illustrates some reconstruction results obtained for regions of an industrial point cloud representing a real petrochemical factory.



(a) Single pipe with noise and occlusions.



(b) Several pipes.



(c) Several pipes.

Fig 10. Some examples of reconstruction results. The input data set contains many occlusions and outliers.

In our experiments, we did not measure the deviation between the original points and the resulting cylinder since P&ID information was used to choose final radii. The precise surface deviation was of little interest in our setting, especially due to presence of huge geometric defects in the input clouds. Careful comparison of our method with other solutions is left for future research because we do not consider our method complete. Moreover, the methods which are reported as fully automatic and precise are not publicly available to make such comparison representative. However, we found interesting to compare our solution with a popular computer vision method RANSAC as reported by [Schnabel et al. 2007]. Our experiments show that RANSAC gives appropriate results for the unambiguously segmented point cloud regions. For industrial point clouds, such segmentation is hardly possible without intensive user interaction. Therefore, fully automatic RANSAC is not an option. The system which employs RANSAC should provide rich interactive means for point cloud segmentation.

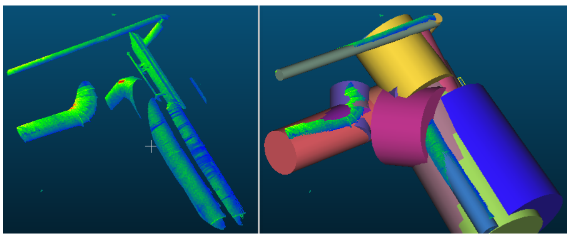


Fig 11. Recognition results using RANSAC method in system [CloudCompare 2017].

Figure 11 illustrates the results of RANSAC method launched for a piece of industrial point cloud (also shown in the Figure 10). The set of extracted cylinders contains many redundant primitives which should be excluded from the result manually.

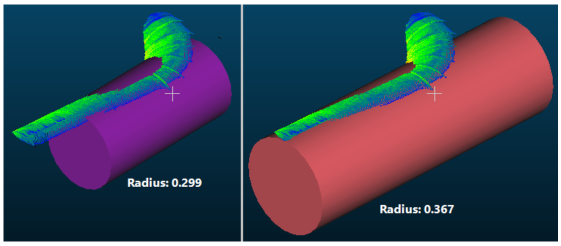


Fig 12. Recognition result using RANSAC method in system [CloudCompare 2017]. One portion of point cloud is recognized in two different ways.

Another observation is related to recognition of an individual primitive. Figure 12 shows that RANSAC method was able to detect two cylinders for a manually captured segment of a real industrial point cloud. The cylinders of different radii give multiple solution in a situation where only one solution is expected. The result of our method is shown in the Figure 13. Since several neighbor bands are used to detect and refine cylindrical cross-sections, our method extracts two radius values and averages them in order to return a single result.

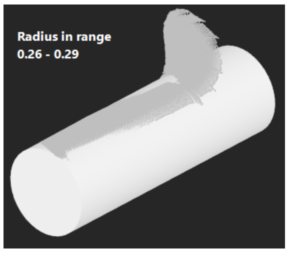


Fig 13. Recognition result using our method.

Another advantage of our method is that it enables *recognition prior to fitting*. Using differential characteristics of a local area, it is possible to decide algorithmically whether consequent fitting makes sense or not. Moreover, since differential characteristics of a point cloud are extracted by general-purpose TPS technique, our method can be extended to recognition and extraction of pipeline connection features.

5. References

[1] Benko, P., Martin, R.R., and Varady, T. 2001. Algorithms for reverse engineering boundary representation models. CAD Computer Aided Design 33, 11, 839–851.

[2] CloudCompare. 3D point cloud and mesh processing software. [online] Available at: http://www.danielgm.net/cc [Accessed 15 August 2017].

[3] Coope, I.D. 1993. Circle fitting by linear and nonlinear least squares. Journal of Optimization Theory and Applications 76.

[4] Hegland, M., Roberts, S., Altas, I., et al. 1997. Finite element thin plate splines for surface fitting. Computational Techniques and Applications: CTAC97 41, 1, 289–296.

[5] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. 1992. Surface reconstruction from unorganized points. ACM SIGGRAPH Computer Graphics 26, 2, 71–78.

[6] Le, T. and Duan, Y. 2017. A primitive-based 3D segmentation algorithm for mechanical CAD models. Computer Aided Geometric Design.

[7] Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., and Mitra, N.J. 2011. GlobFit: Consistently Fitting

Primitives by Discovering Global Relations. ACM Trans. Graph. 30, 4.

[8] Liu, Y.J., Zhang, J. Bin, Hou, J.C., Ren, J.C., and Tang, W.Q. 2013. Cylinder detection in large-scale point cloud of pipeline plant. IEEE Transactions on Visualization and Computer Graphics 19, 10, 1700–1707.

[9] Masuda, H. and Tanaka Ichiro. 2010. As-Built 3D Modeling of Large Facilities Based on Interactive Feature Editing. Computer-Aided Design and Applications 7, 1–10.

[10] Muja, M. and Lowe, D.G. 2014. Scalable Nearest Neighbour Algorithms for High Dimensional Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 11, 2227–2240.

[11] Qiu, R., Zhou, Q.Y., and Neumann, U. 2014. Pipe-run extraction and reconstruction from point clouds. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8691 LNCS, PART 3, 17–30.

[12] Rabbani, T. 2006. Automatic Reconstruction of Industrial Installations Using Point Clouds and Images. Publications on geodesy 62, May, 7401–7410.

[13] Rabbani, T., van den Heuvel, F. a, and Vosselman, G. 2006. Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences – Commission V Symposium “Image Engineering and Vision Metrology” 36, 5, 248–253.

[14] Schnabel, R., Wahl, R., and Klein, R. 2007. Efficient RANSAC for point-cloud shape detection. Computer Graphics Forum 26, 2, 214–226.

[15] Slyadnev, S. 2014. Open CASCADE Technology Overview. [online] Available at: http://isicad.net/articles.php?article_num=17368 [Accessed 3 May 2017].

[16] Slyadnev, S. 2017. Analysis Situs software for B-Rep inspection and prototyping CAD algorithms. [online] Available at: http://www.analysissitus.org [Accessed 3 May 2017].

[17] Son, H., Bosché, F., and Kim, C. 2015. As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey. Advanced Engineering Informatics 29, 2, 172–183.

[18] Son, H. and Kim, C. 2016. Automatic segmentation and 3D modeling of pipelines into constituent parts from laser-scan data of the built environment. Automation in Construction.

[19] Tran, T.T., Cao, V.T., and Laurendeau, D. 2015. Extraction of cylinders and estimation of their parameters from point clouds. Computers and Graphics (Pergamon) 46, 345–357.

About authors

Vadim Turlapov (vadim.turlapov@itmm.unn.ru): professor of Computer Science, Lobachevsky Nizhny Novgorod University.

Sergey Slyadnev (sergey.slyadnev@gmail.com): PhD student, Lobachevsky Nizhny Novgorod University.