

A point cloud decomposition by the 3D level scanning for planes detection



Pavel Chmelar*, Lubos Rejfeek, Ladislav Beran, Martin Dobrovolny

Department of Electrical Engineering, Faculty of Electrical Engineering and Informatics, University of Pardubice, Pardubice, Czech Republic

ARTICLE INFO

Article history:

Received 12 February 2017

Received in revised form

18 September 2017

Accepted 19 September 2017

Keywords:

Point cloud

Plane detection

Level scanning

Component labeling

ABSTRACT

A point cloud represents a set of measurement points. Usually it is a group of points in a defined coordinate space without any information how individual points relates to each other. For a simple shapes and objects description additional methods are needed. In this paper we would like to present a new 3D point cloud scanning method for planes detection. Our developed algorithm includes several image processing methods like the connected component labeling and the shape borders detection which allows computing important plane properties and even detect object shapes. The scanning algorithm is described on a testing example and verified on real measured data. The paper concludes by algorithm properties summarization and recommendations where this method can be used.

© 2017 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

During a 3D range scanning process is created a resulting 3D point cloud. If measurement data are not additionally processed, the measurement result is a simple set of individual points with coordinates in a defined coordinate system. Points covering objects in a surrounding space to describe objects and shapes present in scene, additional detection methods are need to classify individual points into groups with common parameters.

Well-known algorithm for a plane and object detection is the RANSAC. This algorithm finds in several iterations all points which belongs to the searched model. There are many variants and modification (Kurban et al., 2015; Fujiwara et al., 2013; Cupec et al., 2009; Schnabel et al., 2007). For example, Kurban et al. (2015) shows the algorithm ability on data measured by the Microsoft Kinect with combination of Point Cloud Library functions. Cupec et al. (2009) showed an implementation on stereo vision images. An implementation by Schnabel et al. (2007) can detect different shapes like planes, spheres, cylinders and cones.

Planes can be detected by different techniques like Hough transform, which presenting (Borrmann et al., 2011). A next approach showed (Deschaud and Goulette, 2010), their algorithm uses filtered normals and voxel growing and it is able to detect

planned in large and noisy point clouds. Interesting methods evaluation offers (Palnick, 2014), where they comparing different methods to select the best and fast plane detection processes combination for an implementation on a robot.

In this paper we want to use a different method which uses the 3D level scanning. Planes are detected on the base of statistical data representation and its area size. Our method includes using of several image processing algorithms to improve plane detection and get important object's properties for a future processing and classification.

The rest of the paper is organized as follows. Section 2 describes individual 3D level scanning algorithm steps. Section 3 shows algorithm results and practical using of proposed methods, which follows the conclusion in Section 4.

2. Scanning algorithm

This section describes a 3D level scanning algorithm. At the algorithm start it is useful to have some initial characteristic about the input point cloud. Its knowledge can help to set appropriate scanning conditions for achieving the best result. The algorithm uses two main parameters; the level step lv/S and the level range step lv/RS .

Both represents statistical plane properties, which algorithm detects and set the selection level range. These parameters meaning is described in next sections. The input point cloud is organized as row vectors with three dimension columns X, Y and Z. Individual algorithm steps are described on the following 3D point cloud, see Fig. 1.

* Corresponding Author.

Email Address: pavel.chmelar@student.upce.cz (P. Chmelar)

<https://doi.org/10.21833/ijaas.2017.011.019>

2313-626X/© 2017 The Authors. Published by IASE.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

This example is not a real measurement, but for better illustration we set the distance units to the centimeters.

Each point from this point cloud is represented as a vector in the Cartesian coordinate space R^3 . Coordinates of a single point p are defined as (Eq. 1)

$$P = (P_x, P_y, P_z) \quad (1)$$

where p_x , p_y and p_z are coordinates in appropriate axis. For points rotations and transformations, the rotation matrixes in the 3D space (2), (3) and (4) are needed, described in paper (Beran et al., 2015).

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2)$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (3)$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\delta) & \sin(\gamma) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

symbols α , β and γ are rotation angles in corresponding axes X , Y and Z .

2.1. Scanning dimension and range selection

The first step is selection of an appropriate scanning dimension. The choice of X , Y or Z dimension is arbitrary, because there is an influence only on the scanning direction. Usually appropriate dimension is selected according scanning purpose. From the testing point cloud data nature it is selected the dimension Z in our example.

The scanning range is a distance range, where individual levels are analyzed. Typically, the range PC_{rng} (5) from the input point cloud is estimated as maximum PC_{rngH} and minimum PC_{rngL} point's position in the appropriate dimension.

$$PC_{m,g} = (PC_{m,gL}, PC_{m,gH}). \quad (5)$$

The selection of different range depends on level decomposition purpose and a what result is expected from a scanning.

2.2. Level step estimation

The scanning process starts by the first detection range selection. The value PC_{rngL} is selected as a low detection range DR_L . The high range DR_H is calculated as

$$DR_H = DR_L + lvls \cdot lvIRS. \quad (6)$$

This selection (6) ensures the suitable range for the first level estimation. The parameter step level $lvls$ characterizes the acceptable amount of noise or points standard deviation; acceptable as a plane. The parameter step level range $lvIRS$ is used as a range, where the level is estimated. The minimal value of

$lvIRS$ should be at least 1 and recommended is about 3.

The detection level dL is estimated as the point cloud data mean value in the selected dimension, which meets with the condition

$$dL = \mu(PC(:, :, PC(:, :, 3)) \geq (dL = lvls) \&\& PC(:, :, 3) \leq (dL + lvls)). \quad (7)$$

The symbol colon in the Eq. 7 means all points in appropriate dimension. The number 3 is the Z dimension.

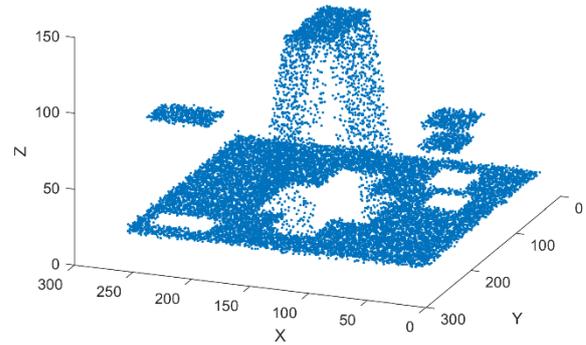


Fig. 1: The input point cloud

Fig. 2 gives a one scanning level step illustration with marked important values DR_H and DR_L . The point cloud data equals to the first scanning level of the Z dimension in this example. The level estimation borders DR_L (8) and DR_H (6) are marked by gray planes. The estimated detection level dL by Eq. 7 is marked as the red plane.

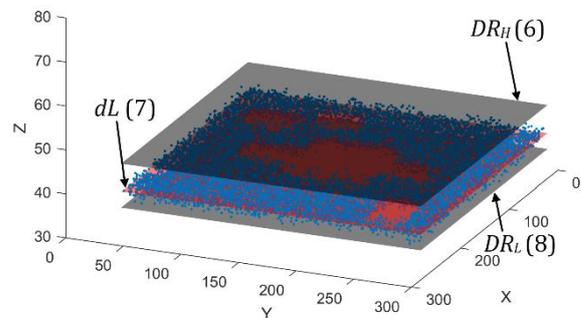


Fig. 2: The level step

For the next algorithm step the value DR_L is selected as

$$DR_L = dL + lvls. \quad (8)$$

You can noticed, that the red level in Fig. 2 is closer to the low border DR_L . The way how level estimation borders DR are set according (6) and (8) during the algorithm processing and proper $lvls$, $lvIRS$ parameter settings for a plane detection with desired standard deviation ensures the right detection level estimation dL in the highest data concentration in an appropriate scanning dimension. Firstly the detection range DR is selected by using both parameters $lvls$ and $lvIRS$, than the mean data

value dL is estimated in scanning dimension. Finally point cloud data are extracted by using estimated mean value dL and the level step $lvlS$ parameter (7). The higher range value for the next step is selected again according (6). The algorithm ends, when the scanning level dL reach to the higher range limit PC_{rngH} .

2.3. Data statistic properties and level image

The main algorithm substance is already mentioned, but this section moves closer to the statistical data representation and point cloud data connection with image processing methods, to achieve more useful information.

When a level dL is estimated by Eq. 7 during one algorithm step, other statistical data are calculated for all dimensions; μ - mean value; σ^2 - variance; σ - standard deviation and \hat{x} - data mode. These statistical data can help with choosing desired planes with specific properties.

Very useful feature is the image construction from a plane. Once the image is created there is information about a plane area and plane borders. Area estimation only from points in point cloud is difficult task. For image construction it is necessary the quantization parameter qD . Choosing the wrong parameter leads to incorrect results. The parameter qD allows getting information about one-pixel physical size in a point cloud image.

Point cloud data positions are quantized into a plane image. When the dimension Z is used in this example, dimension X a Y determine positions on a plane image. The pixel's intensity in appropriate positions are set as the true. The left top plane image corner physical coordinates $I_{phys}(x, y)$ are also retain for the future using.

The next illustration, in Fig. 3, shows the image example created from "bottom" level of the input data point cloud.

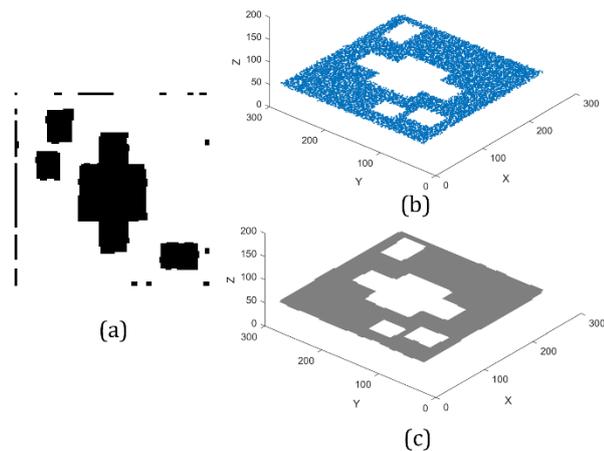


Fig. 3: The level image: (a) – plane image, (b) – plane image in physical dimensions, (c) – original point cloud data

If the parameter qD has small value or the point's density is lower, there can be some undesired pixel

holes in a compact image. For this purpose, it is recommended to use morphological operations like image close to fill undesired gaps and image open to suppress small outlying points.

A resulting image plane can consist of several areas, see Fig. 4. This can happen when two independent planes are at the same level or there are some perpendicular planes or planes with a different rake angle. This situation is solved by the using connected component labeling with eight connections, to cover all connected pixels. In this case for all connected components are recalculated statistical data. Fig. 5 illustrates different statistics data for individual components.



Fig. 4: A presence of more image components

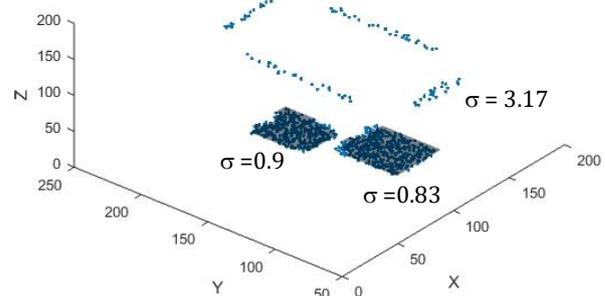


Fig. 5: Image components statistics

Point cloud points for one component are extracted by component analysis. As we denoted individual image pixels have physical dimension qD and image origin position is available. The physical pixel position p_{phys} is

$$P_{phys}(x_{phys}, y_{phys}) = (I_{phys}(1) + x \cdot qD, I_{phys}(2) + y \cdot qD) \tag{9}$$

where x, y are pixel coordinates in a plane image and variables marked by $phys$ are physical dimensions. Desired points are extracted from a component physical range. All statistical properties; the area size and position can be used for planes classification.

2.4. Planes decomposition

Previous sections describing one level scan through a selected dimension. For all planes

segmentation it is necessary to rotate input point cloud according the next illustration.

Rotation angles in the horizontal plane and in the vertical plane are marked in Fig. 6. The horizontal rotation is 360 degrees always and for the vertical plane the angle of 180 degrees is sufficient, because this rotation covers all planes in space without a needles plane repeating in the second half of the rotation sphere.

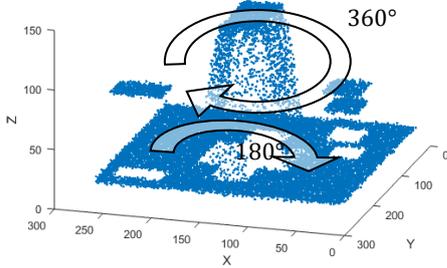


Fig. 6: The point cloud rotation

Individual points are marked by indexes during the scanning process, which labels scanned planes p_x . The label criterion is the maximum plane area p_{xA} .

$$P_x = \max(P_{xA}). \tag{10}$$

If there is found a bigger area for a point during the scanning process, this point is labeled, that belongs to the bigger plane area.

To provide these rotations in the same way how it is depicted in Fig. 6 the two rotations are needed. The first rotation is in scanning dimension (horizontal rotation h) R_h and the second is around one of two left dimensions (vertical rotation v) R_v . The second rotation is various in dimensions, because the 360 degrees rotation is used in scanning dimension and only the algorithm direction is changed.

We can combine rotation matrixes by the multiplication and create the final rotation matrix (11) like using (Dunn and Parberry, 2015). The matrix combination order depends on the real rotations order. Rotation matrixes are square, which makes possible to combine them in any order.

$$R = R_h \cdot R_v. \tag{11}$$

Our example uses the Z scanning dimension, the rotation matrix around Z axis (4) is used and as the vertical, the rotation around Y axis (3) is selected. In the next section are evaluated results and shown some practical examples.

3. Results evaluation and examples of using

The decomposition result of the input point cloud, which illustrating individual algorithm steps is shown in Fig. 7.

The 9 planes are decomposed from approximately 12 500 input points. The area and the rotation angle characterize each detected plane. All planes data are summarized in the Table 1, where the selected important statistical data denoted Section 2.3 is included too.

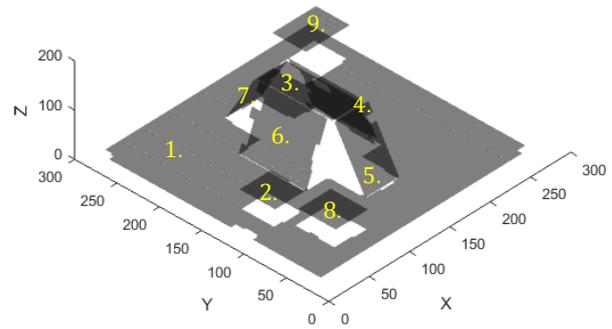


Fig. 7: The decomposition result

One 3D level scan step of illustrated point cloud takes about 200ms in Matlab. To decompose the whole point cloud as is depicted in Fig. 7 takes about 65 000 scanning level step which is significantly time consuming. The one level scan processing time can be significantly decreased by using suitable hardware resources or using effective and image processing libraries. Time crucial parts are morphological operations.

From this drawback it is taken the advantage that the algorithm is well parametrized. Sometimes can be useful to detect only planes in specific orientations and conditions. Usually can be required to get planes only in a specific dimension and if the decomposing point cloud is well oriented only three algorithm steps are needed.

Table 1: Planes decomposition table

Pl.	Area [m ²]	R(h; v) [°]	I _{phys} (x; y) [cm]	dL [cm]	σ (x; y; z) [cm]
1	5,09	0; 0	10; 19	20	79.33;
					78.96; 0.89
2	0,167	0; 0	35; 96	59	9.62;
					11.79; 0.9
3	0,324	0; 0	122; 111	137	25.15;
					0.9
4	0,93	0; 79	-107; 114	183	41.46;
					23.05; 1.76
5	0,493	90; 73	-71; -136	232	35.35;
					11.67; 0.96
6	0,973	180; 78	-163; -195	-86	40.35;
					22.94; 1.28
7	0,479	270; 73	-164; -120	-62	38.37;
					11.40; 1.29
8	0,18	0; 0	49; 39	59	10.69;
					12.18; 0.83
9	0,235	0; 0	202; 220	90	15.14;
					10.64; 0.89

Table 1 describes all detected planes from Fig. 7. The first column are plane numbers from Fig. 7; the second column is plane areas; the column marked by R shows vertical v and horizontal h rotations, where individual planes were detected; the next column contains physical origin of plane images; the following column is estimated detection level dL and

the last include a mean values of points, which belongs to individual planes.

According rotation angles from the Table 1. it is obvious, that five planes were found during the first 3D level scan. Negative values in the table are caused by angle rotations. Standard deviation for the Z dimension in the last column shows the proof of concept; all detected planes have this value lower than used parameter lv/S . For a plane reconstruction as in Fig. 7, the rotation angles $R(v,h)$ are needed and the original plane point cloud. Data from this point cloud are rotated in $R(v,h)$; then the plane image is created. Extracted plane borders from a plane image with using inverse form of matrix (11) are used for a plane visualization as illustrates Fig. 7. Rotation matrixes are orthogonal and an inverse matrix can be done by a simple matrix transposition.

One important plane property is its area and this can be used for example in exploring space in front of the scanning device. Fig. 8 shows the range view in front of the autonomous scanning system introduced in (Chmelar and Dobrovlny, 2013).

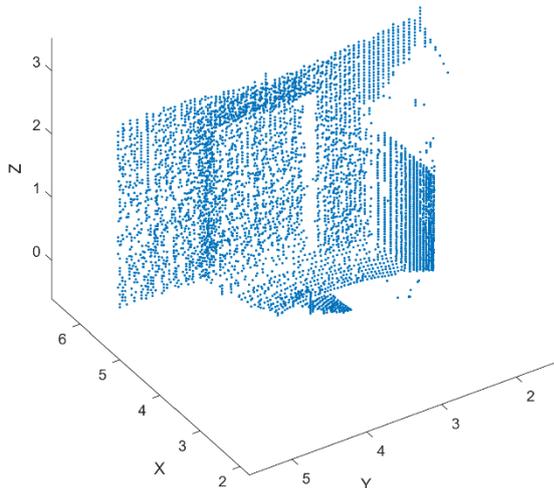


Fig. 8: The rangefinder range view point cloud

This range view was used in our previous work, where we introduced the depth map construction from a 3D point cloud (Chmelar et al., 2016).

From this view we need for example information about doors area and wall area. The rangefinder view is oriented in X direction. Only one 3D level step is required to get this information. The point cloud area which covers the wall, see Fig. 9, is 9.16 m² and the door's area in Fig. 10 is 4.08 m². By this information we can easily analyzed the point cloud properties and an object dimension present in a scene.

For illustration the parameters $lv/S = 2$, $lv/RS = 3$ and $qD = 2$ were used in this example. New algorithm possibilities are wide and it can be used in different situation. For the planes and objects detection can be used other effective and fast detection methods, which used the mathematic models for its detection, which were introduced in Section 1. The main paper contribution is the object point cloud connection

with the well-known image processing methods to get important object properties.

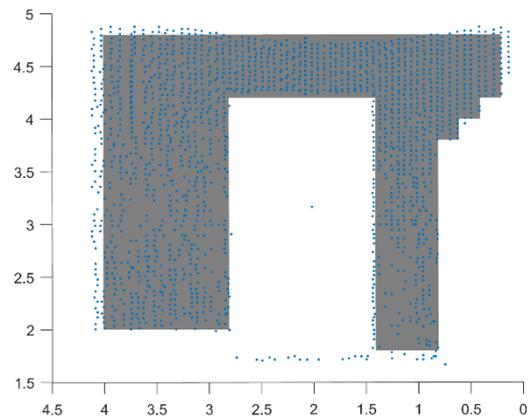


Fig. 9: The wall's area

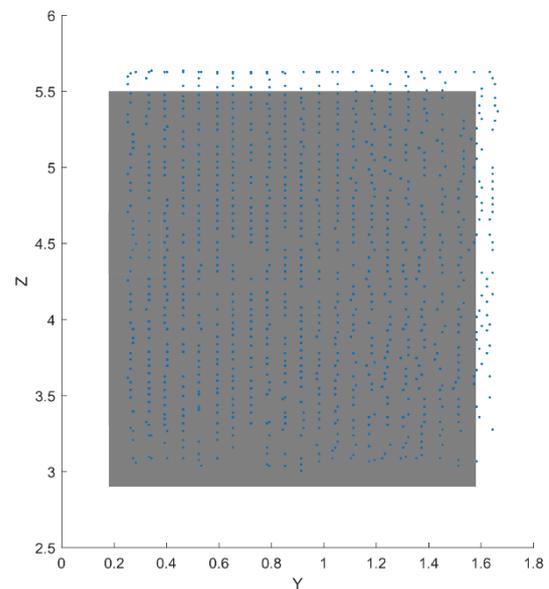


Fig. 10: The door's area

4. Conclusion

The paper describes the point cloud decomposition algorithm by using the 3D level scanning. The planes detection is its primary goal. The advanced statistical plane properties are available for all individual planes. Results evaluation shows its detection capability, advantages and few disadvantages. The main disadvantage can be time processing of unorganized point cloud, especially its orientation, in case of all presented planes detection. This lack it is possible to compensate by using suitable image processing libraries or by appropriate hardware resources. Usually the detection is done only in desired angles and if the point cloud orientation is known the whole scanning process is simplified.

The algorithm connects point cloud data with often used image processing methods. This offers to easily observe very important object properties, which can be used for future processing. For

detection individual objects or shapes can be used sophisticated mathematical methods and once an object points are extracted, the described image processing methods in this paper enables getting important object properties. Even if a 3D volumetric object like spheres, humans and others are present, we can suppress depth information and follows object shapes, which are useful features for object recognition in many image processing areas. Our future research will be focused in this area.

Acknowledgment

The research was supported by the Internal Grant Agency of University of Pardubice, the project No. SGS_2016_022.

References

- Beran L, Chmelar P, and Rejcek L (2015). Navigation of robotics platform using monocular visual odometry. In the 25th International Conference on Radioelektronika (RADIOELEKTRONIKA), IEEE: 213-216. <https://doi.org/10.1109/RADIOELEK.2015.7129012>
- Borrmann D, Elseberg J, Lingemann K, and Nüchter A (2011). The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2): 1-13.
- Chmelar P and Dobrovolny M (2013). The fusion of ultrasonic and optical measurement devices for autonomous mapping. In the 23rd International Conference on Radioelektronika (RADIOELEKTRONIKA), IEEE: 292-296. <https://doi.org/10.1109/RadioElek.2013.6530934>
- Chmelar P, Beran L, and Rejcek L (2016). The depth map construction from a 3D point cloud. *MATEC Web of Conferences (ICMIE 2016)*, EDP Sciences, 75: 1-6. <https://doi.org/10.1051/mateconf/20167503005>
- Cupec R, Grbic R, Nyarko EK, Sabo K, and Scitovski R (2009). Detection of planar surfaces based on RANSAC and LAD plane fitting. In the 4th European Conference on Mobile Robots (ECMR'09), Dubrovnik, Croatia: 37-42.
- Deschaud JE and Goulette F (2010). A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In the Conference of 3D Processing, Visualization and Transmission Conference (3DPVT'10), Paris, France.
- Dunn F and Parberry I (2011). *3D math primer for graphics and game development*. CRC Press, Boca Raton, USA.
- Fujiwara T, Kamegawa T, and Gofuku A (2013). Plane detection to improve 3D scanning speed using RANSAC algorithm. In the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA'13), IEEE: 1863-1869. <https://doi.org/10.1109/ICIEA.2013.6566671>
- Kurban R, Skuka F, and Bozpolat H (2015). Plane segmentation of kinect point clouds using RANSAC. In the 7th International Conference on Information Technology, Amman, Jordan: 545-551. <https://doi.org/10.15849/icit.2015.0098>
- Palnick J (2014). Plane detection and segmentation for DARPA robotics challenge. Ph.D. Dissertation, Worcester Polytechnic Institute, Worcester, USA.
- Schnabel R, Wahl R, and Klein R (2007). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, Blackwell Publishing Ltd, 26(2): 214-226.