Proceedings of the Fifth IIEEJ International Workshop
on Image Electronics and Visual Computing 2017
Da Nang, Vietnam, February 28- March 3, 2017

IMPROVED 3D PLANES DETECTION USING SCALED DIFFERENCE OF NORMALS

Jaime SANDOVAL†
†Shinshu University

Kazuma UENISHI †
†Shinshu University

Munetoshi IWAKIRI‡
‡National Defense Academy of Japan

Kiyoshi TANAKA†
†Shinshu University

ABSTRACT

3D planes detection is an important task that has numerous applications in urban environments. However, current methods do not deal appropriately with the noise and quantization artifacts of low-cost sensors. In this paper, we present the Scaled Difference of Normals, a points filter that addresses these issues and is implemented on top of the Fast and Deterministic Planes Detection Based on Hough Transform. We evaluated its precision by comparing the detected planes coefficients with semi-automatically generated ground truth data and confirmed that when compared to state of the art methods, the proposed method is fast and has superior precision even in the presence of high noise levels, quantization artifacts and several variations in the points distribution caused by registration.

1. INTRODUCTION

3D point clouds processing and pattern recognition is now an important topic in computer vision and robotics. With the advent of low-cost 3D sensors and its integration into a variety of devices such as drones, indoors robots and mobile devices, the development of new algorithms for 3D pattern recognition has gained strong attention.

A 3D point cloud is a collection of points in $\mathbb{R}^3$ space, they are generated by 3D sensors which acquire objects surface into a distribution of points, it is noted that sometimes they may contain other information such as color, intensity and point normals. However, for the sake of generalization we will assume that this information is not available.

Several applications scenarios include indoors and a variety of urban environments, in which most of the scenes and objects are composed of numerous planar surfaces, such as walls, floors, streets, stairs or buildings, among others. Therefore, 3D planes detection is a critical task for these applications.

However, the state of the art algorithms are visually imprecise or too slow to be used in real time, some of them require previous knowledge of the Point Clouds such as surface normals or noise level.

In this work we present a novel filtering method for 3D planes detection, by selecting planar points using Difference of Normals (DoN) at different scales we optimized point clouds for use in planes detection algorithms.

We build this filtering method on top of a previous work[2] in order to increase its accuracy in both noisy and noiseless environments.

2. PREVIOUS METHODS

2.1. HOUGH TRANSFORM

In [2], planes are detected by dividing the point cloud in voxels and discarded if their points do not posses planar properties based on Difference of Centroids (DoC) and eigenvalues. Consequently, the plane spanned by the voxel is transformed into Hough Space and stored in a low-memory sparse accumulator.

However, as PCA is used to calculate the voxel plane, it is prone to noise, outliers, and quantization artifacts of low-cost sensors. An approach to solve this issue is to increase the voxel size, nonetheless, this can prevent the algorithm to detect smaller planes.

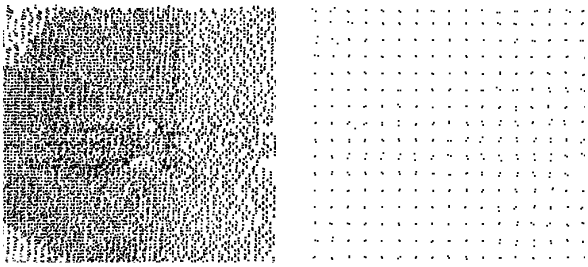
By observation, we noticed that RANSAC performs well on the coplanar voxels, as these typically contain a few number of inliers, therefore, we made use of pseudo-random RANSAC plane fitting to calculate the coefficients.

At the time RANSAC is used, the planarity of the points is already known and the probability of finding plane outliers is low, hence, we set the RANSAC probability to 0.9. Moreover, in order to keep the method deterministic, the random seed is set to a fixed number (1234).

Additionally, the distance threshold (in [m]) is set accordingly to the smallest eigenvalue,

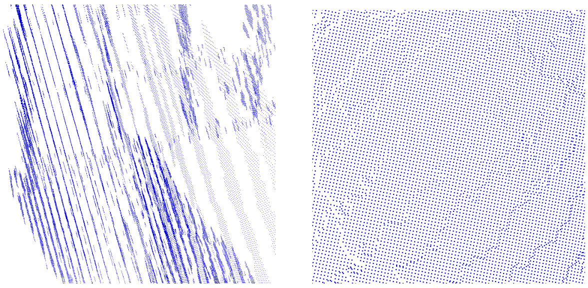
$$T_h = \max(\frac{\sqrt{\lambda_1}}{10}, 0.001)\ [\mathrm{m}], \qquad (1)$$

where $\lambda_i \le \lambda_{i+1}, i \in \{1, 2, 3\}$, and the RANSAC threshold



(a) Raw and noisy. (b) After Voxel Grid Filter.

Figure 2: Noisy planar section before and after Voxel Grid Filter.



(a) Side view. (b) Front view.

Figure 1: Front and side view of a wall sensed by a simulation of a Kinect sensor with quantization artifacts.

is defined as

$$R_{T_h} = \frac{Th}{3} \text{ [m]}. \quad (2)$$

2.2. DIFFERENCE OF NORMALS

Difference of Normals (DoN)[1] is a multi-scale operator for segmentation and object recognition, it consists on subtracting two normal vectors $\mathbf{n_1}$ and $\mathbf{n_2}$ at different radius r_1 and r_2 and calculating its L2 norm:

$$DoN = \left\| \frac{\mathbf{n_1} - \mathbf{n_1}}{2} \right\|_2 . \quad (3)$$

Nonetheless, to use this formula directly, the normals have to be oriented towards a viewpoint, hence, to avoid this issue the absolute value of the cosine distance $|\text{COS}_\text{d}(\mathbf{n_1}, \mathbf{n_2})|$ is used as the DoN value:

$$DoN = |\text{COS}_\text{d}(\mathbf{n_1}, \mathbf{n_2})| = \left| \frac{\mathbf{n_1} \cdot \mathbf{n_2}}{\|\mathbf{n_1}\| \|\mathbf{n_2}\|} \right| \in [0, 1]. \quad (4)$$

The outcome of **Eq. 4** is a measure of similarity between two vectors in the range $[0, 1]$, where 0 indicates orthogonality and 1 indicates they are pointing to the same or the opposite direction, that is, the orientation of the calculated normals does not affect this measurement.

In spite of the multi-scale property of DoN which makes it stronger to Gaussian Noise, when acquiring point clouds from low-cost sensors, we observed that DoN becomes inaccurate due to the quantization artifacts shown in **Fig. 1**, hence giving negative influence to the calculations.

3. PROPOSED METHOD

3.1. SCALED DON

In order to alleviate the issues of DoN we proposed to look for the DoN at lower scales of the point cloud, since in these scales, the adverse influence of noise and quantization artifacts are greatly reduced, additionally, planar surfaces tend to be more stable and well defined.

To scale down the point cloud we use a downsampling algorithm available in PCL library[6]: the voxel grid filter. This filter creates an octree of the point cloud and calculates the centroid of each leaf voxel, these centroids become the new points instead of the original, thus softening or eliminating the adverse influence of high noise scanners.

In **Fig. 2a**, We can observe the planar section of a point cloud with simulated Kinect noise using Blensor 1.0.17[5]. In **Fig. 2b** the results after applying the voxel grid filter are shown, notice that the planar structure is clearer, however, the points tend to become far apart.

Therefore, by assuming that planar surfaces become more stable at lower scales, we designed a method that decides which point to pick according to the DoN values at different scales (i.e. DoN > 0.9), hence constructing a point cloud with only planar points.

In a nutshell, the proposed method scales down the point cloud repeatedly and in each scale, DoN is calculated for all its points using the Eq. 4. If the DoN is planar at every scale, then the point lies on a local plane and is selected for later use in Planes Detection.

Although not being optimal at the beginning, this method is adapted to [2] as a pre-filtering to test the precision. Further speed improvements may include the use of the normals calculated in DoN although this does not affect accuracy.

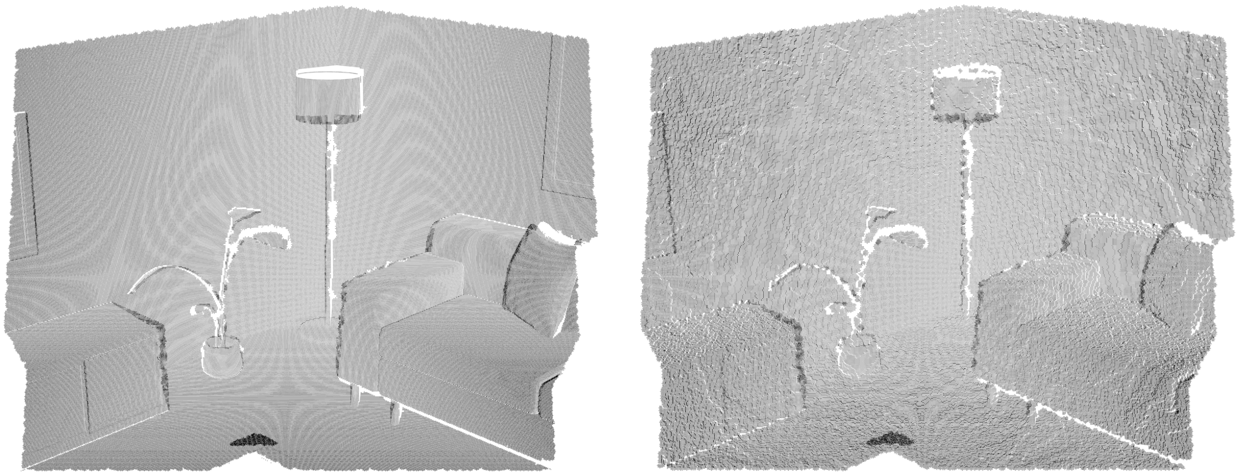
Notice that normals calculation method is critical to its accuracy. The most common method is PCA based on the covariance matrix as described in [4], which is known to be prone to several types of noise, however, the study of normals calculation is out of the scope of this paper and will remain for future work.

4. EVALUATION

4.1. DATASET

Table 1: Dataset detailed information

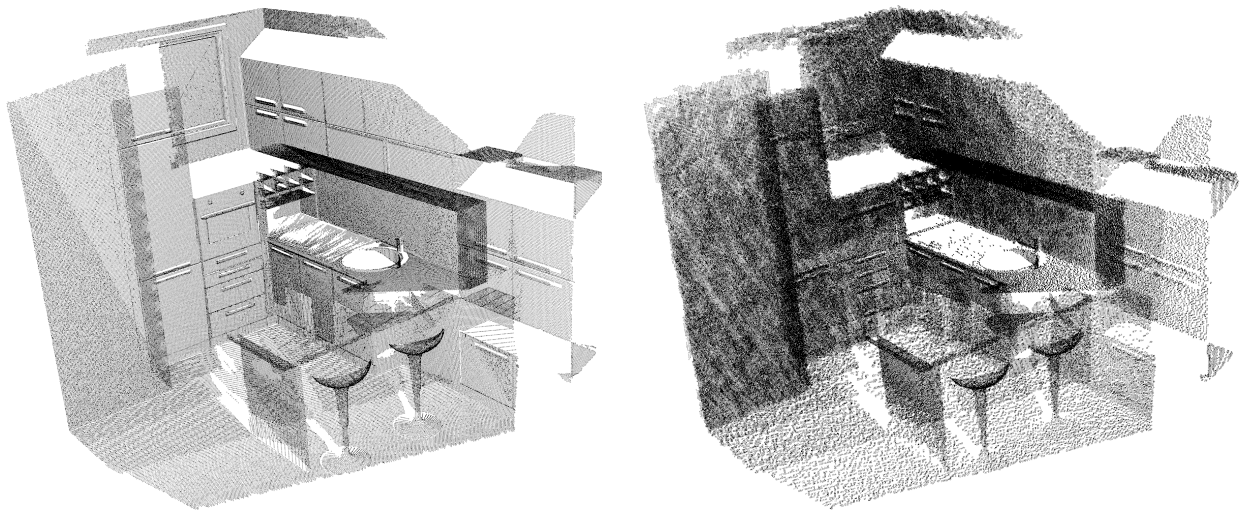
Name	**Mesh resolution[mm]**	**Points[#]**	**AABB volume [m^3]**
Room	5.08	307,200	19.29
Room (noisy)	5.05	307,200	19.20
Kitchen	6.75	245,928	29.04
Kitchen (noisy)	6.80	494,507	31.02



(a) Room

(b) Room (noisy)

Figure 3: Noisy and noiseless Room point cloud obtained by Kinect simulation



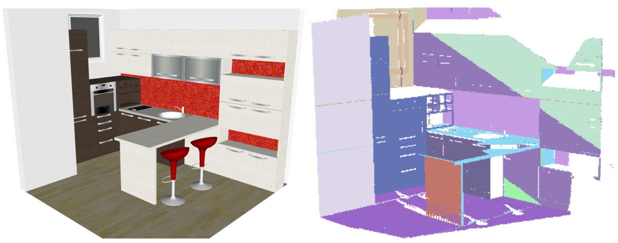
(a) Kitchen

(b) Kitchen (noisy)

Figure 4: Noisy and noiseless Kitchen point cloud obtained by the concatenation of 10 Kinect scans simulation

Algorithm 1 Precision evaluation

```
procedure EVALUATE(Planes_gt, Planes_det)
    m ← emptyList
    while not empty Planes_det do
        l ← emptyList
        for all Planes_gt do
            b ←bestMatch(Planes_det)                ▷ Eq. 4
            l.add(b)
        match ← best(l)
        m.add(match)
        deleteMatched(Planes_det, m)
    recreate(Planes_det)
    n_d ←countMatched(m, Planes_det)
    for all Planes_gt ∉ m do
        p_e ←bestMatch(Planes_det)                  ▷ Eq. 4
        m.add(match)
    A_cc = 0
    for all Planes_gt ∈ m do
        Planes_det ←detectedList(Plane_gt)
        Err_gt = error(Plane_gt, Planes_det)        ▷ Eq. 5
        A_cc = A_cc + Err_gt
    A_cc = A_cc / size(m)                           ▷ overall accuracy
    return A_cc, n_d
```



(a) Ground truth model. (b) Ground truth planes.

Figure 5: Ground truth data for the kitchen model.

We made use of indoor 3D models of a kitchen and a room, then we simulated a Kinect scanner using Blensor, which can generate both clean and realistic noisy point clouds. The room point clouds shown in **Fig. 3a** and **Fig. 3b** were created with a single Kinect scanning, while the Kitchen point clouds (**Fig. 4a** and **Fig. 4b**) were created by simulating a noisy and noiseless registration. This was done in Blensor by moving the camera 10 times through a path, at each step, both noiseless and noisy Kinect scans were obtained in world coordinates. Finally, the point cloud was concatenated to simulate a perfect registration.

To quantitatively test the accuracy of a planes detection method, a ground truth database of planes is necessary. For this purpose, we calculated plane candidates using the noiseless point clouds and a modification of the RANSAC plane detection approach in [3] (hereinafter referred as CFRANSAC). The current approach is a coarse-to-fine plane detection, however, small planes are sometimes not found or incorrectly detected by RANSAC. Therefore, to avoid this issue, an ultra-fine detection step was added to its algorithm using the same parameters and thresholds of the fine segmentation step, this advanced algorithm will be referred as UFRANSAC.

After generating the planes, they are manually selected to get a cleaner result. Thereupon their coefficients are saved as the ground truth data. In **Fig. 5**, it is shown the projection of the ground truth planes inliers. Therefore, we can visually confirm the correctness of the ground truth data.

4.2. EVALUATION METHOD

In order to evaluate the precision of the methods, two metrics are used; the overall accuracy A_{cc} and the correct planes detected n_d, the algorithm used is described in **Alg. 1**. It consists of a clustering-like algorithm in which the ground truth data define each cluster, then, the detected planes are matched once into their best cluster.

Once the matching is finished, the number of correctly detected planes (n_d) is calculated. In this step undetected ground truth planes will not be associated with detected planes, therefore, to apply a penalty to this case, a second matching of these ground truth planes is performed on the whole set of detected planes.

Consequently, we can measure the detection error of each ground truth plane (Err_{gt}) by adding up the error between itself and its m number of associated detected planes coefficients $\mathbf{P_{det_n}}$,

$$Err_{gt} = \sum_{n=1}^{m} \left(1 - \mathrm{COS_d}(\mathbf{P_{gt}}, \mathbf{P_{det_n}})\right), \tag{5}$$

then we define A_{cc} as the average of the additive error Err_{gt} of the ground truth planes P_{gt} with their associated detected planes, $\mathbf{P_{det_n}}$, i.e.,

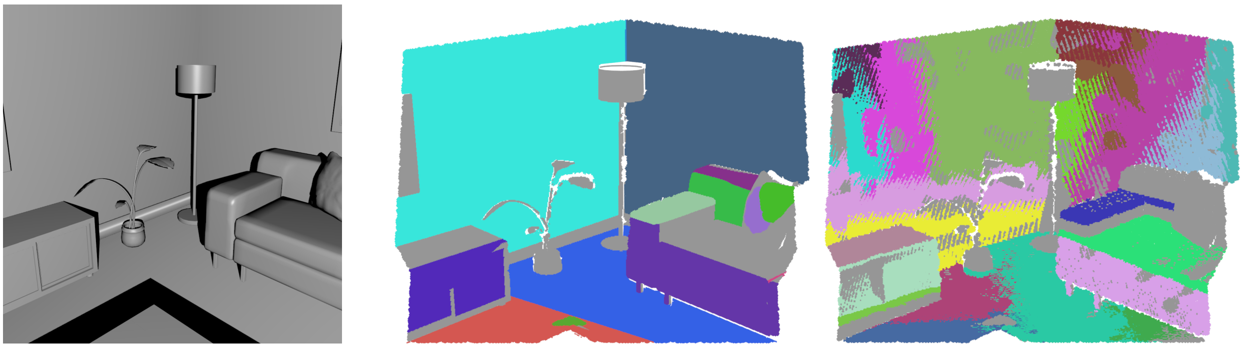
$$A_{cc} = \frac{\sum_{n=1}^{T_{gt}} Err_{gt_n}}{T_{gt}} \tag{6}$$

where T_{gt} is the number of ground truth planes.

Additionally, the processing time of each method is evaluated using wall time. For the case of nondeterministic algorithms, the average of 50 executions is shown in the results for both precision and processing time metrics.

Table 2: Room detection results, 12 ground truth planes, best values are highlighted

		CFRANSAC	UFRANSAC	RHT	FDHT	Proposed
Without noise	A_{cc}	0.0569	**0.0562**	0.0889	0.0564	0.0828
	n_d	9	**12**	9	9	9
	T[s]	6.9511	8.0847	26.599	**0.4659**	0.5442
With noise	A_{cc}	0.2048	0.2004	0.2782	0.3848	**0.0759**
	n_d	5	5	8	9	**12**
	T[s]	11.7268	12.6784	20.667	0.4382	**0.4293**

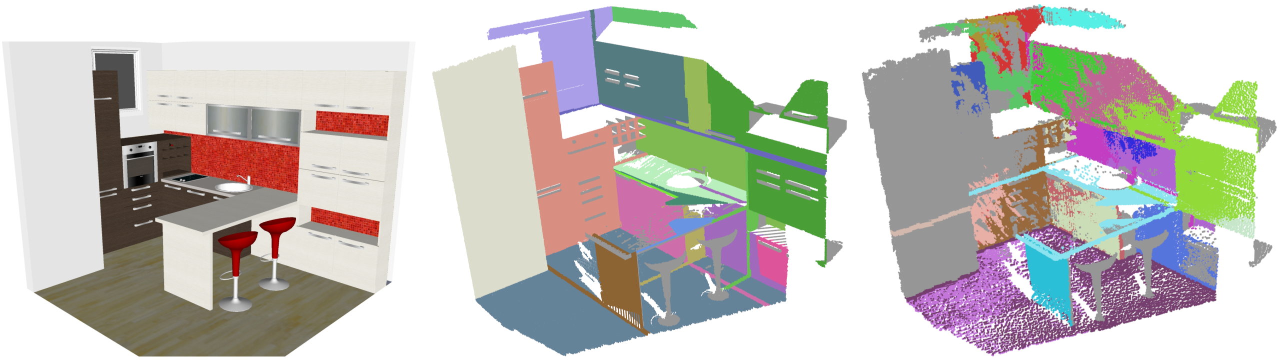


(a) Ground truth (b) Noiseless detection (c) Noisy detection

Figure 6: Room detection results of the proposed method when using a noiseless and noisy scan

Table 3: Kitchen detection results, 16 ground truth planes, best values are highlighted

		CFRANSAC	UFRANSAC	RHT	FDHT	Proposed
Without noise	A_{cc}	0.0122	**0.0003**	0.634	0.0416	0.0575
	n_d	14	15	10	14	**16**
	T[s]	6.7115	7.36042	6.34165	**0.3918**	0.4611
With noise	A_{cc}	3.1112	3.1294	0.9796	1.3503	**0.3528**
	n_d	7	8	10	11	**12**
	T[s]	29.4502	32.0889	28.3123	**0.8149**	1.302



(a) Ground truth (b) Noiseless detection (c) Noisy detection

Figure 7: Kitchen point cloud detection results of the proposed method when using a noiseless and noisy scan

4.3. EVALUATED METHODS

For each method, the overall accuracy, the number of correctly detected planes, and the processing time are evaluated.

Regarding the RANSAC family both CFRANSAC and its improved version UFRANSAC are used. For the Hough Transform based methods we have the Randomized Hough Transform (RHT)[7], and the Fast and Deterministic Hough Transform (FDHT)[2]. Moreover, the proposed method, as it is based on the later, it falls into the Hough Transform based category as well.

4.4. RESULTS

The results of the experiments are shown in **Tab. 2** and **Tab. 3**. In both tables the noisy and noiseless point cloud result is evaluated, the Err row is the overall accuracy A_{cc} in Alg. 1, the pad (#) row shows the number of correct planes detected, and the T[s] row shows the processing time in seconds. For CFRANSAC, UFRANSAC and RHT, the values are the average of 50 executions

In Tab. 2, within the noiseless scenario, the proposed method had better precision but did not detect all the planes, in the noisy scenario the proposed method clearly outperforms, it is noteworthy that the processing time is comparable to the FDHT while the others are orders of magnitude slower.

In Tab. 3, within the noiseless scenario, UFRANSAC had better overall accuracy because it was tuned to work with this kind of dataset, where the point cloud is clean and contains several small planes. However, the proposed method found all the ground truth planes while being fast, comparable to the FDHT processing time. In the noisy scenario the proposed method had superior accuracy and number of detected planes, while being not as fast as FDHT but around 21 times faster than RHT.

Visual results of the proposed method are shown in **Fig. 6** and **Fig. 7**. Best viewed in color. For the room point clouds, the ground truth model is **Fig. 6a**, the results when using the noiseless point cloud is **Fig. 6b**, which shows that the most prominent planar sections were successfully detected and segmented. In the noisy plane detection results of **Fig. 6c**, the multi-coloring of big planar regions reflects a small angular offset of the detected planes due to noisy quantization.

This quantization artifact is a well-detached part of a planar surface; therefore, when a noise reduction or upsampling mechanism is applied, a valley is generated instead of a planar points distribution.

The kitchen noisy model (**Fig. 7c**) is a particular case. One of the undetected walls is composed of quantized and noisy registered scans, hence making it one of the worst cases of noise that can be generated artificially. Therefore, we can confirm that the noiseless detection (**Fig. 7b**) can estimate all the planes, while in the worst case of noise, it can detect around 75% of the ground truth planes with high accuracy and speed.

5. CONCLUSIONS AND DISCUSSION

We proposed a method to improve the detection of 3D Planes in high noise scenarios. The experiments show that this method has superior precision in noisy scenarios, while having fast speed and good precision in noiseless data.

PCA is being used to calculate the points normals, however, as stated in [1], it is prone to noise and therefore, future work will focus on the precision of normals calculation. Moreover, to gain speed, we think that the point normals generated by DoN can be integrated into the Plane Detection process.

REFERENCES

[1] Yani IOANNOU, et al., Difference of normals as a multi-scale operator in unorganized point clouds. In 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission. IEEE, 2012. pp. 501-508.

[2] Jaime SANDOVAL, Kazuma UENISHI, Munetoshi IWAKIRI and Kiyoshi TANAKA, A Fast and Deterministic Plane Detection Method in Point Clouds based on Hough Transform, In Proceedings of VC/GCAD 2016.

[3] Kazuma UENISHI, Jaime SANDOVAL, Munetoshi IWAKIRI and Kiyoshi TANAKA, Feature Description using Planar Surfaces for 3D Virtual Keypoint, In Proceedings of VC/GCAD 2016 (in Japanese).

[4] Yizhou YU, Surface reconstruction from unorganized points using self-organizing neural networks, In IEEE Visualization, 1999, pp. 61-64.

[5] Michael GSCHWANDTNER, et al., BlenSor: blender sensor simulation toolbox, In International Symposium on Visual Computing, Springer Berlin Heidelberg, 2011, pp. 199-208.

[6] RUSU, Radu Bogdan, Steve COUSINS, 3d is here: Point cloud library (pcl), In Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011, pp. 1-4.

[7] Dorit BORRMANN, et al., The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design, 3D Research, 2011, vol. 2, no 2, pp. 1-13.