Proceedings of the 2016 IEEE
International Conference on Robotics and Biomimetics
Qingdao, China, December 3-7, 2016

A Slope Location and Orientation Estimation Method Based on 3D LiDAR Suitable for Quadruped Robots*

Xiangrui Meng, Chao Zhou, Zhiqiang Cao, Leijie Zhang, Xilong Liu, and Shuo Wang

Abstract—**3D LiDAR is widely used for mobile robots environment perception nowadays, which has high precision of distance measurement. In this paper, a calibration method which is suitable for Velodyne VLP-16 is firstly introduced. And then we utilize statistical filtering for isolated points removal. After statistical filtering, a slope location and orientation estimation method is presented based on PROSAC algorithm, which can achieve plane extraction from the 3D point cloud data. Experiments are conducted to testify the performance of the presented method.**

I. Introduction

Quadruped robots receive much attentions, and they have better adaptability in unstructured and complex environments with stones, bumps, holes or slopes. A challenge is its environment perception for different environments. 3D LiDAR is suitable because it has high precision of distance measurement. Slope is such a kind of terrain which should be detected in advance. Different angles and different orientations of slopes lead to different path planning and gait adjustments. Similar researches on 3D LiDAR and mobile robots have been conducted [1-3].

As the development of sensors based on 3D distance measurement such as 3D LiDAR, a growing amount of researches have been done. Douillard *et al.* [4] presented a set of segmentation methods for various densities of 3D point clouds. It first provides empirical evidence of the benefit of ground extraction prior to object segmentation in the context of dense data. The authors compared the simplicity, accuracy and computation times of these algorithms on several sets of hand labeled data using two metrics. Jaebum [5] proposed a SLAM method using Rao-Blackwellized particle filters (RBPFs) based on a hybrid map which combines a grid map and a feature map. The experiments prove that the approach works well in outdoor environments and the uncertainty of a predicted vehicle position decrease drastically. Fioraio *et al.* [6] built a real-time visual SLAM based on point cloud data. The method performs generalized ICP on two frames in a very short time. Additional information from visual point features can be considered so that robustness and increased accuracy will be achieved. Shackleton *et al.* [7] introduced an approach to detect and track people in real-time using a 360° LiDAR sensor. Hernández *et al.* [8] illustrated a segmentation method for urban ground modeling using point cloud data from 3D LiDAR. The detection is achieved by Top-Hat of hole filling algorithm of range images and feature extraction is fulfilled with detected connected components. This method classifies the artifacts on the ground into four categories in the experiments with SVM algorithm.

And there are also some researches on slope and plane detection using various sensors. Lv *et al.* [9] presented a method to detect a slope by using two-dimensional LRF (laser range finder) for robot navigation tasks. Oehler *et al.* [10] proposed a multi-resolution approach for planar components segmentation with 3D point cloud data. The experiments are conducted with good quality and efficiency. Unlike other researches which use sensors based on distance or vision, Suwanratchatamanee *et al.* [11] used tactile sensing system to detect a slope for humanoid robot. They simplify the walking control method by using a real-time tactile feedback instead of a complex ZMP calculations. The experiments show that the humanoid robot can walk on different uneven grounds including upward and downward slopes.

In this paper, we address a slope location and orientation estimation method with a 3D LiDAR sensor. This paper is organized as follows. Section II introduces the calibration method for 3D LiDAR, Velodyne VLP-16. The statistical filtering algorithm is then described in Section III. Also, slope location and orientation estimation method based on PROSAC algorithm is discussed in Section III. In Section IV, experiments are conducted. Section V concludes this paper.

II. Calibration for 3D Lidar

The 3D LiDAR is mounted on the head of the quadruped robot. Coordinate systems including laser coordinate system and world coordinate system are constructed, which are shown in Fig. 1. The former coordinate system $O_LX_LY_LZ_L$ is established with its origin $O_L(x_L, y_L, z_L)$ being the bottom center of the 3D LiDAR sensor, and the latter coordinate system $O_WX_WY_WZ_W$ is established with its origin $O_W(x_W, y_W, z_W)$ being the point just below $O_L(x_L,y_L,z_L)$ a certain length.

Calibration of 3D LiDAR consists of internal parameters calibration and external parameters calibration. Internal parameters calibration aims to find out the parameters on the

*This work is supported in part by the National Natural Science Foundation of China under Grants 61273352, 61233014, 61421004, and in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2015AA042201, and in part by the national defense basic research program under Grant B132011××××, and in part by the Beijing Natural Science Foundation under Grants 4161002.

Xiangrui Meng, Chao Zhou, Zhiqiang Cao, Leijie Zhang, and Shuo Wang are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. (E-mail: {mengxiangrui2014, chao.zhou, zhiqiang.cao, zhangleijie2014, shuo.wang}@ia.ac.cn)

Xilong Liu is with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. Xilong Liu is the corresponding author. (E-mail: xilong.liu@ia.ac.cn).

These authors are also with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408, China.

978-1-5090-4364-4/16/$31.00 ©2016 IEEE

transformation from measured points to the points described in the laser coordinate system. And the external parameters calibration is used to establish the transformation from laser coordinate system to world coordinate system.

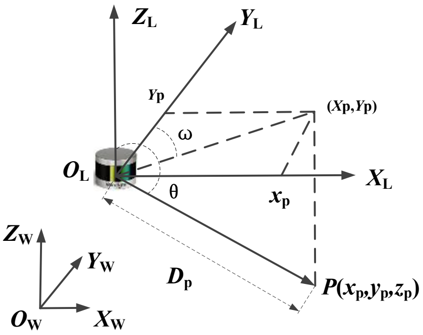


Figure 1. Laser coordinate system and world coordinate system

A. Internal Parameters Calibration

For 3D LiDAR, measurement values including distances, vertical angles and horizontal angles can be acquired. However, the measurement values returned from laser head cannot be viewed as data in the laser coordinate system. For one reason, in view of the mechanical structure and installation error, the optical center and the mechanical center cannot be the same. In this case, internal parameters calibration is necessary. Generally speaking, this work has been done by the manufacturer of the 3D LiDAR and internal parameters are given, so the detailed procedures will not be discussed in this paper.

B. External Parameters Calibration

External parameters calibration means coordinate system transformation from laser coordinate system to world coordinate system. Rotation matrix *R* and translation matrix *T* has to be set. In this paper, translation matrix *T* can be simplified due to the fact that the world coordinate system $O_wX_wY_wZ_w$ is just below the laser coordinate system.

First, the rotation matrix *R* is estimated, which is related to roll angle, pitch angle, and drift angle. In order to illustrate the estimation of *R*, Euler Angles [12] are introduced, as shown in Fig. 2. We consider a ground plane that is flat, large enough, and clear of obstacles for calibration. This plane's equation can be assumed as $Ax+By+Cz+D=0$. Obviously, the normal vector of the plane is (*A*, *B*, *C*). In this case, we can get a coordinate system $O_RX_RY_RZ_R$ with its origin being the point $O_R(x_R,y_R,z_R)$, which can be seen as a temporary coordinate system for the transformation without considering R_d and *T*. And X_R-axis, Y_R-axis, and Z_R-axis of the coordinate system are:

$$\begin{aligned}
\overrightarrow{Z_R} &= (A,B,C) \\
\overrightarrow{X_R} &= (C,0,-A) \\
\overrightarrow{Y_R} &= \overrightarrow{Z_R} \times \overrightarrow{X_R} \\
N &= (B,-A,0)
\end{aligned} \tag{1}$$

where *N* is intersection line of plane $X_LO_LY_L$ and $X_RO_RY_R$.

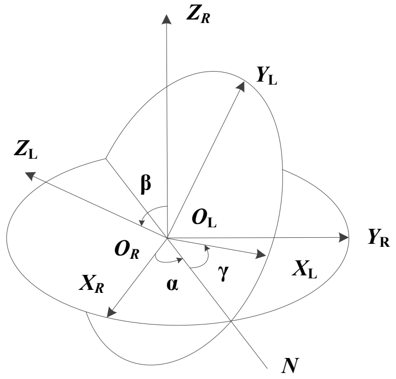


Figure 2. Euler angles

The Euler angles can be deduced by the estimation of X_R-axis, Y_R-axis, and Z_R-axis, which are shown as equation (2).

$$\begin{aligned}
\alpha &= \arccos\frac{\overrightarrow{X_R}\,\overrightarrow{N}}{\left|\overrightarrow{X_R}\right|\left|\overrightarrow{N}\right|} \\
\beta &= \arccos\frac{\overrightarrow{Z_R}\,\overrightarrow{Z_L}}{\left|\overrightarrow{Z_R}\right|\left|\overrightarrow{Z_L}\right|} \\
\gamma &= \arccos\frac{\overrightarrow{N}\,\overrightarrow{X_L}}{\left|\overrightarrow{N}\right|\left|\overrightarrow{X_L}\right|}
\end{aligned} \tag{2}$$

where α is the angle between X_R-axis and *N*-axis, β is the angle between Z_L-axis and Z_R-axis, and γ is the angle between X_L-axis and *N*-axis.

It should be noted that each of these angles α, β, and γ have a direction. And the rotation matrix R_r related to roll angle and pitch angle is given by

$$R_r = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

In addition to R_r, the parameters of R_d related to drift angle are determined as the following steps. Several poles are put in the environment, which is illustrated in Fig. 3. And a temporary coordinate system $O_DX_DY_DZ_D$ is established without considering translation matrix *T*, with its origin being the point $O_D(x_D,y_D,z_D)$. For a specific point *P* on a pole, $P_i(X_i, Y_i, Z_i)$ is the coordinate of *P* in $O_DX_DY_DZ_D$, and $p_i(x_i, y_i, z_i)$ is the coordinate of *P* in $O_RX_RY_RZ_R$.

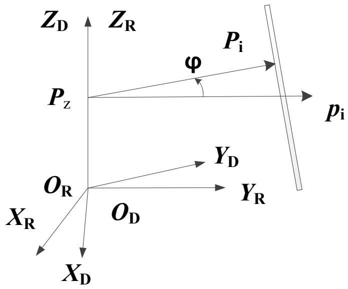


Figure 3. Drift angle estimation

φ is the drift angle as shown in Fig. 3, and calculated by

$$\varphi = \arccos \frac{\overrightarrow{P_z p_i} \overrightarrow{P_z P_i}}{\left|\overrightarrow{P_z p_i}\right| \left|\overrightarrow{P_z P_i}\right|} \tag{4}$$

where P_zP_i and P_zp_i are vectors with related points in the coordinate system $O_DX_DY_DZ_D$ and the coordinate system $O_RX_RY_RZ_R$, respectively.

It should be noted that the drift angle φ have a direction. R_d can be described as:

$$R_d = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Based on the estimation above, the rotation matrix R is calculated as follows:

$$R = R_d R_r \tag{6}$$

Then the translation matrix T is estimated. As the world coordinate system is just below the laser coordinate system with a length of h vertically, T can be estimated by equation (7), which is the transformation matrix from the temporary coordinate system $O_DX_DY_DZ_D$ to world coordinate system $O_wX_wY_wZ_w$.

$$T = \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \tag{7}$$

Finally, the external parameters calibration are determined. And the transformation is given by

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = R \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + T \tag{8}$$

where (x_w, y_w, z_w) and (x_l, y_l, z_l) are coordinates in the world coordinate system and laser coordinate system, respectively.

III. Slope Location and Orientation Estimation

A. Isolated Points Removal Using Statistical Filtering

The point cloud dataset often contains uneven data because of the measurement or system error, etc. In order to remove isolated points, some researchers utilize point cloud features such as normal vector or rate of curvature at each point. However, the computation is much high and new wrong values may be generated. In this paper, statistical method is utilized for isolated points removal [13]. Generally speaking, statistical analysis is done for each point based on the distance distribution of points nearby. For a specific point pt, m is the number of neighborhood points, if the distribution of its neighbors is a Gaussian distribution, the mean distance μ and standard deviation σ are calculated. The points which are not within the range will be eliminated. This process is shown as equation (9).

$$S_{pt} = \{pt_i \mid pt_{ng} \in S_{ng}, D_{pt} \le \mu \pm \kappa \times \sigma\} \tag{9}$$

where S_{pt} is the dataset of clustered points, pt_i is an element of S_{pt}, pt_{ng} is an neighborhood point of pt_i, S_{ng} is the dataset of pt_{ng}, D_{pt} is the distance between pt_i and pt_{ng}, and κ is multiplier of standard deviation.

B. Slope Location and Orientation Estimation Based on PROSAC Algorithm

Unlike RANSAC algorithm [14], PROSAC algorithm [15] will sort the data by some rules defined by users. This paper assume that the slope is close to a plane, and we define the plane in the world coordinate system, as shown in equation (10).

$$\rho = x'\cos\alpha' + y'\cos\beta' + z'\cos\gamma' \tag{10}$$

where ρ is the distance between the plane and the original point O_w, α', β', γ' are angles between the plane and three axes, respectively, (x', y', z') is the coordinate of a point in the plane.

Apart from certain number of points that fit the plane, the point dataset also has large amount of outliers, which should be excluded. First, the algorithm selects three points randomly from the point cloud dataset to set up an initial plane equation. Second, the algorithm will check whether other points in the dataset fit the plane or not. If the distance between the point and the plane is within the threshold, which defines the maximum deviation, the point is considered as an inlier. And then a consensus set is built by the set of inliers. If the point is beyond the threshold, it will be considered as an outlier. Afterwards, the plane is estimated using all the points from the consensus set. Third, the inliers will be sorted by distances between the points and the fitting plane. The points that are close to the fitting plane will have higher probability to be selected to form an initial three-point group for next iteration. Finally, if there are sufficient points classified into inliers or the maximum iteration time is reached, the fitting plane will be considered as a good one. The plane parameters including the normal vector u_{xoy} of the plane are then obtained.

On the basis of normal vector u_{xoy}, location and orientation of the slope can be estimated. Robots need to know the distance and orientation between the slope and themselves before they step onto the slope. In order to illustrate the location of slope, this paper defines the starting line segment of the slope by the following equations:

$$\begin{aligned} & p_1(x_1, y_1, z_1) = \{q \mid \max(q_x + q_y), q \in L_{start}\} \\ & p_2(x_2, y_2, z_2) = \{q \mid \min(q_x + q_y), q \in L_{start}\} \\ & L_{start} = S_\Phi \cap P_\Pi \end{aligned} \tag{11}$$

where $p_1(x_1, y_1, z_1)$, $p_2(x_2, y_2, z_2)$ are the end points of the starting line segment, and L_{start} means the set of starting line of the slope, q is a point in L_{start}, q_x, q_y are x, y coordinates of q, respectively, S_Φ is the set of slope points, and P_Π is the set of ground points.

It should be noted that L_{start} may not be a line measured by the 3D LiDAR, but just a computed line. In this way, this paper defines a 9-duple to illustrate location of the slope, as shown in equation (12).

$$\Lambda_{\mathrm{L}} = (x_1, y_1, z_1, x_2, y_2, z_2, l_{\mathrm{x}}, l_{\mathrm{y}}, l_{\mathrm{z}}) \tag{12}$$

where $x_1, y_1, z_1, x_2, y_2, z_2$ are coordinates of p_1, p_2, respectively, l=(l_{x}, l_{y}, l_{z}) is the direction vector of L_{start}.

If we define that z coordinates of the ground points are 0, the location of the slope Λ_{L} can be simplified as:

$$\Lambda_{\mathrm{LS}} = (x_1, y_1, x_2, y_2, l_{\mathrm{x}}, l_{\mathrm{y}}) \tag{13}$$

When the moving direction of the robot is perpendicular to L_{start}, the slope orientation is defined as the projective of moving direction on the slope. The orientation of the slope is given by

$$\vec{\psi} \cdot \vec{l} = 0, \vec{\psi} \in S_{\Phi} \tag{14}$$

where Ψ is the slope orientation vector.

If the robot is already on the slope, the stepping angle becomes important. The stepping angle can be defined as follows. If the moving direction of the robot is perpendicular to L_{start}, it is equal to slope angle; and if the moving direction is not perpendicular to L_{start}, the stepping angle δ_{s} can be calculated with the help of Λ_{L} or Λ_{LS} as shown in equation (15).

$$\begin{gathered} \delta_{\mathrm{s}} = \arctan \frac{\tan \eta \tan \delta}{\sqrt{\tan^2 \eta + \tan^2 \delta + 1}} \\ \eta = \arccos \frac{\vec{s} \cdot \vec{l}}{|s||l|}, s = (s_{\mathrm{x}}, s_{\mathrm{y}}, s_{\mathrm{z}}), l = (l_{\mathrm{x}}, l_{\mathrm{y}}, l_{\mathrm{z}}) \end{gathered} \tag{15}$$

where δ is the angle of slope, η is the intersection angle between moving direction and l, s is the vector of moving direction.

IV. Experiments

Velodyne VLP-16 is adopted in this paper as the 3D LiDAR. Unlike general single-layer LiDAR, VLP-16 supports 16 channels, a 360° horizontal field of view and a 30° vertical field of view. Distances, azimuth angles and intensity values are data measured by the 3D LiDAR. The following experiments are fulfilled under a visualization platform built by visual studio 2013 with PCL 1.8.0 [16]. In the following experiments, the horizontal field of view is set to 180°. The original point cloud which illustrates the environment is shown in Fig. 4, where different lines represents different channels, and the color reflects the distance between the LiDAR and that point. The experiments conducted in this section consist 7 different slope angles from 10.5° to 23.8°.

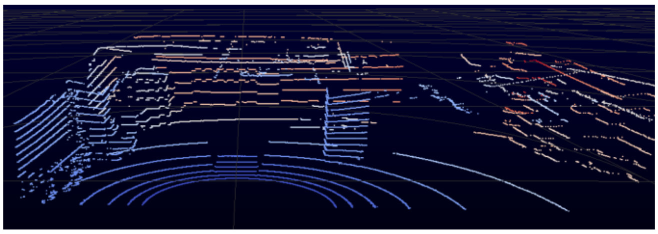
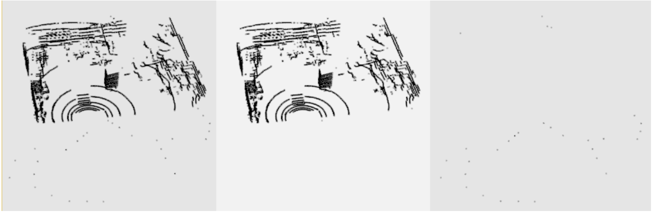


Figure 4. Original point cloud data, which describes the environment of experiments.

A. Statistical Filtering

With statistical filtering, isolated points which are far from the point cloud clusters are eliminated. In this experiment, m=30, κ=2.0, the number of original points is 21600. Statistical filtering result is illustrated in Fig. 5, original point cloud is shown in Fig. 5 (a), filtered point cloud is shown in Fig. 5 (b), and the isolated points are shown in Fig. 5 (c). One can see that after statistical filtering, the left points are 21539 with a rate of 99.72%. It is obvious that points which are not close to clusters have been filtered.



(a) (b) (c)

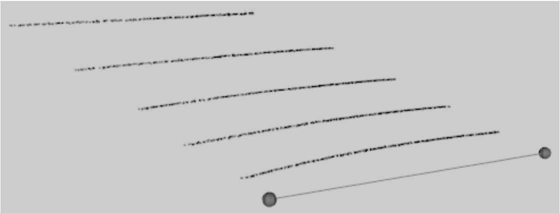
Figure 5. Statistical filtering. (a) shows the original point cloud, (b) shows the filtered point cloud, and (c) shows the isolated points.

B. Slope Location and Orientation Estimation

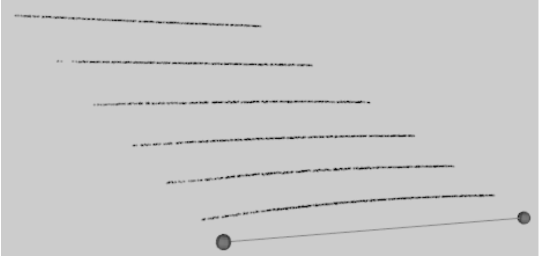
The starting segment of the slope is estimated in Tab. 1, where *m_Angle* column describes the ground truths of slopes measured by the inclinometer, *m_LineVector* column reflects the starting line estimation, *m_EndPoints* column depicts the end points corresponding to the starting line. The slope starting line segment is shown in Fig. 6, where a slope of 13.5° is illustrated in Fig.6 (a), and a slope of 19.7° is illustrated in Fig. 6 (b). And they are illustrated in the bottom of the pictures.

TABLE I. Starting Line Segment Estimation

No.	m_Angle	m_LineVector	m_EndPoints
1	10.5°	(0.0042,0.1887,0.0000)	(1.9330,-0.6047,-0.0009) (1.9584,0.5429,-0.0009)
2	13.5°	(0.0073,0.2323,0.0000)	(1.9432,-0.6240,-0.0008) (1.9800,0.5432,-0.0008)
3	15.3°	(0.0072,0.2591,0.0000)	(1.9425,-0.6193,-0.0008) (1.9744,0.5361,-0.0008)
4	17.4°	(0.0038,0.2796,0.0000)	(1.9588,-0.6247,-0.0015) (1.9743,0.5259,-0.0015)
5	19.7°	(0.0038,0.3237,0.0000)	(1.9840,-0.6181,-0.0026) (1.9974,0.5178,-0.0026)
6	22.0°	(0.0055,0.3587,0.0000)	(1.9875,-0.6074,-0.0006) (2.0048,0.5195,-0.0006)
7	23.8°	(0.0069,0.4014,0.0000)	(2.0000,-0.6150,-0.0013) (2.0195,0.5277,-0.0013)



(a)



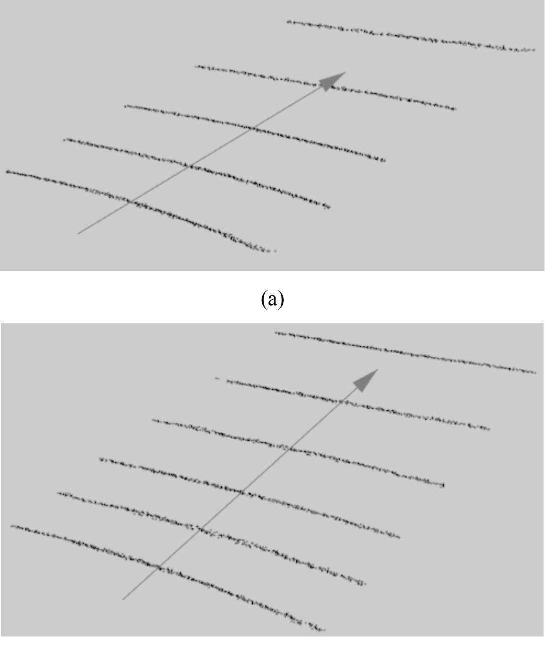
(b)

Figure 6. Slope location estimation, where the line segment in the bottom of the picture means L_{start} of the slope. (a) *m_Angle*=13.5°, (b) *m_Angle*=19.7°

Slope orientation indicates a moving direction on the slope which is perpendicular to L_{start}. The slope orientation vector is estimated in Tab.2, where *m_OrientationVector* reflects the slope orientation estimation. The orientation vectors are illustrated in Fig. 7, where a slope of 13.5° is illustrated in Fig. 7 (a), and a slope of 19.7° is illustrated in Fig. 7 (b). And the arrows mean the orientations of the slopes.

TABLE 2. SLOPE ORIENTATION VECTOR ESTIMATION

No.	m_Angle	m_OrientationVector
1	10.5°	(0.1853,-0.0041,0.0356)
2	13.5°	(0.2259,-0.0071,0.0540)
3	15.3°	(0.2502,-0.0069,0.0672)
4	17.4°	(0.2685,-0.0036,0.0782)
5	19.7°	(0.3063,-0.0036,0.1048)
6	22.0°	(0.3349,-0.0052,0.1287)
7	23.8°	(0.3676,-0.0063,0.1612)



(a)

(b)

Figure 7. Slope orientation estimation, where the arrow means the orientation of the slope. (a) *m_Angle*=13.5°, (b) *m_Angle*=19.7°

V. CONCLUSIONS

It is necessary for quadruped robots to detect slopes in the environments, because different slopes lead to different path planning and gait adjustment. In this paper, a slope location and orientation estimation is presented based on PROSAC with a 3D LiDAR. Internal and external parameters calibration for Velodyne VLP-16 are also discussed. And statistical filtering is utilized for isolated points removal. Experiments show that the location and orientation of the slopes in the specific environments are estimated well.

REFERENCES

[1] G. Atanacio-Jiménez, J. González-Barbosa, J.B. Hurtado-Ramos, F.J. Ornelas-Rodríguez, H. Jiménez-Hernández, T. García-Ramirez, and R. González-Barbosa, "LIDAR velodyne HDL-64E calibration using pattern planes," *International Journal of Advanced Robotic Systems*, 8(5), pp. 70-82, 2011

[2] E. Shang, X. An, M. Shi, D. Meng, and J. Li, "An Efficient Calibration Approach for Arbitrary Equipped 3-D LiDAR Based on an Orthogonal Normal Vector Pair," *Journal of Intelligent & Robotic Systems*, 79(1), pp. 21-36, 2014

[3] C.L. Glennie, A. Kusari, and A. Facchin, "Calibration and Stability Analysis of the VLP-16 Laser Scanner," in *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016, pp. 55-60

[4] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LIDAR point clouds," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2798-2805

[5] C. Jaebum, "Hybrid Map-based SLAM using a Velodyne Laser Scanner," in *International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 3082-3087

[6] N. Fioraio, and K. Konolige, "Realtime Visual and Point Cloud SLAM," in *Proceeding of the RGB-D workshop on advanced reasoning with depth cameras at robotics: Science and Systems*, 2011

[7] J. Shackleton, B. Vanvoorst, and J. Hesch, "Tracking People with a 360-Degree Lidar," in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010, pp. 420-426

[8] J. Hernández, and B. Marcotegui, "Point cloud segmentation towards urban ground modeling," in *Joint Urban Remote Sensing Event*, 2009

[9] J. Lv, Y. Kobayashi, Y. Hoshino, and T. Emaru, "Slope detection based on orthogonal assumption," in *IEEE/SICE International Symposium on System Integration*, 2012, pp. 61-66

[10] B. Oehler, J. Stueckler, J. Welle, D. Schulz, and S. Behnke, "Efficient multi-resolution plane segmentation of 3d point clouds," in *International Conference on Intelligent Robotics and Applications*, 2011, pp. 145-156

[11] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, "Walking on the slopes with tactile sensing system for humanoid robot," in *International Conference on Control Automation and Systems*, 2010, pp. 350-355

[12] Wikipedia, Euler angles, [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles

[13] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, 56(11), pp. 927-941, 2008

[14] M.A. Fischler, and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 24(6), pp. 381-395, 1981

[15] O. Chum, and J. Matas, "Matching with PROSAC-progressive sample consensus," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 220-226

[16] R.B. Rusu, and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1-4