IEIE Transactions on Smart Processing and Computing, vol. 4, no. 6, December 2015
http://dx.doi.org/10.5573/IEIESPC.2015.4.6.403

Supervoxel-based Staircase Detection from Range Data

Ki-Won Oh and Kang-Sun Choi*

Interdisciplinary Program in Creative Engineering, KOREATECH / Cheonan, South Korea
{o6789, ks.choi}@koreatech.ac.kr

***** Corresponding Author: Kang-Sun Choi

Received December 14, 2015; Accepted December 22, 2015; Published December 31, 2015

** Short Paper*

Abstract: In this paper, we propose a supervoxel clustering–based staircase extraction algorithm to obtain poses and dimensions of staircases from a point cloud. In order to effectively reduce the candidate points and accelerate supervoxel clustering, large planes in the scene, such as walls, floors, and ceilings, are eliminated while scanning the environment. Next, staircase candidates with small planes are initially estimated using supervoxel clustering. Then, parameter values for the staircases are refined, and higher staircases that remain undetected due to occlusion are predicted and generated virtually. Experimental results show that staircases are detected accurately and predicted successfully.

Keywords: Staircase, Plane detection, Supervoxel clustering, Prediction, Point cloud

1. Introduction

Mobile robots have become more commonplace in commercial and industrial settings. Mobile robots that perform human tasks move around their environment. Robots have been used in hospitals to move materials for many years. Recently, mobile robots have begun operating in multi-floor indoor environments where staircases are common features. However, mobile robots often encounter difficulties in identifying and navigating staircases. The ability to safely identify and navigate staircases is important for the prevention of serious injury to people and damage to the mobile robots [1]. In order to perform human tasks in multi-floor indoor environments, it is important to have a detection method for stairs that are present when moving to a target location.

To detect staircases, a method using random sample consensus (RANSAC) was proposed [1]. This method uses the Microsoft Kinect to find staircases in the same interval. But the results from detection include points that correspond to walls connected to staircases. Therefore, elimination of points for other components is needed. To solve this problem, it is necessary for the points of walls, ceilings, and floors to be eliminated. To eliminate these objects, a method of plane detection is needed.

An et al. proposed a fast plane-detection method that detects planes during data gathering [3]. While scanning the environment, points are converted into line segments and clustered onto planes. It is effective for calculations to be processed during point cloud acquisition. However, this method easily fails to detect planes for small areas.

In this paper, we propose a supervoxel clustering-based staircase extraction algorithm to accurately obtain poses and dimensions of staircases. For stairs that are undetected due to occlusion, a method to predict undetected stairs is also presented.

2. Proposed Method

Fig. 1 shows a flow chart of our proposed algorithm. At first, we eliminate points of large planes, like walls, floors and ceilings, to reduce the processing time of the method and eliminate unrelated points that exist. In this process, we apply a plane detection method, which is processed per scan [3]. For the remaining points, we extract staircase parameters using a supervoxel clustering–based staircase extraction algorithm, and create virtual steps by predicting ascent staircases by using the extracted parameters. In Section 1, we reviewed a detection method [3], and in the

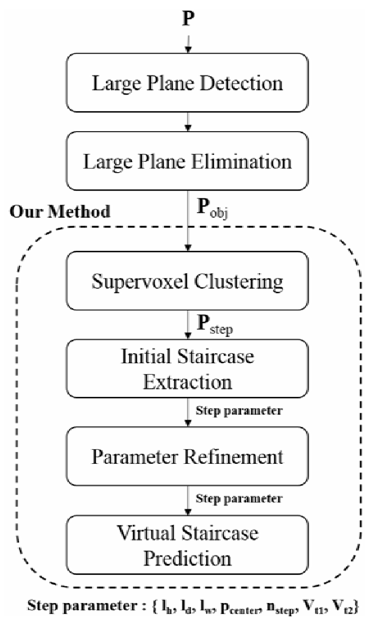

Fig. 1. Flow chart of the proposed stair-detection method.

next section, we introduce the proposed stair detection algorithm.

2.1 Large Plane Detection

2.1.1 Line Segment Extraction

In each scan, points p_i can be described by range and bearing from light detection and ranging (LIDAR), so (ρ_i, θ_i) is obtained. p_i can be converted to 2D Cartesian coordinates ($\rho_i\cos\theta_i$, $\rho_i\sin\theta_i$). To extract line segments, a breakpoint representing discontinuities in the sequence of scan points is detected per scan. A breakpoint detection algorithm examines the distance between two consecutive points (p_i, p_{i-1}) using the following condition:

$$\| p_i - p_{i-1} \| < D_o, \tag{1}$$

where D_0 is a threshold. If Eq. (1) is satisfied, p_i and p_{i-1} are considered to be in the same cluster. Otherwise, p_i and p_{i-1} are considered breakpoints and are put into different clusters. The distance between two consecutive points is proportional to ρ_{i-1}. So, D_0 should be dependent on ρ_{i-1}. An intuitive method for D_0 was proposed [4]:

$$D_o = \rho_{i-1} \cdot \sin(\Delta\phi) / \sin(\lambda_u - \Delta\phi) + 3\sigma_d, \tag{2}$$

where φ is the sensor resolution, and λu is chosen on the basis of user experience. The term $3\sigma d$ is added in order to account for noise associated with ρi when $\mathbf{p}i$ is close to the sensor frame origin. Quantitative determination of λ is an unresolved problem, but in mobile robot applications $\lambda u = 10^\circ$ has been reported to be satisfactory.

After finding the breakpoint process, clusters include a polygonal line. To include only the line segment, a splitting process that divides the polygonal line into line segments is performed. To do this, iterative end-point fitting (IEPF) was adopted. This method has been used in many vision and robotics applications. First, the line segment li is generated by the end of two points of a cluster. Next, the distances h_j between the line and points included in a cluster are calculated. The maximum value of h_j is compared with threshold h_0. If $h_{j(max)}$ is bigger, the cluster is divided into two clusters at the point corresponding to $h_{j(max)}$.

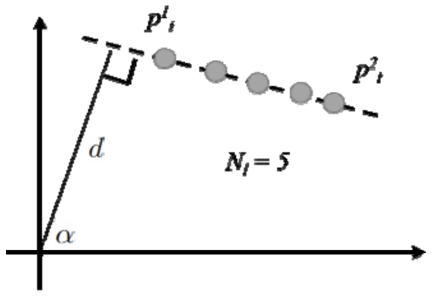

Fig. 2. Parameters of a line.

Each cluster after the splitting process can be expressed by line parameters, as shown in Fig. 2. p^1_i, p^2_i is two end points of a line, and N_l is the number of points included in the line segment. After extraction of the line segment, a merging process is performed to combine short line fragments by checking the following condition:

$$| \alpha_i - \alpha_j | < \alpha_0 \wedge \| p_i^2 - p_j^1 \| < d_0. \tag{3}$$

Also, lines that are not effective in the detection of a plane are eliminated. These are lines that are found to be less than threshold L_{min}, or that have too few scan points.

$$\| p_i^2 - p_j^1 \| < L_{\min} \vee n_i < N_e. \tag{4}$$

The line segment extracted in each scan step is represented by two-dimensional (2D) coordinates, and it has to be converted into 3D coordinates. Dimensions of two 2D end points of a line segment are converted into 3D points represented by $W = (u, v)$ by using scan information of the end point and the angle of actuator ψ.

$$\begin{aligned} W_x &= \rho\cos\theta, \\ W_y &= \rho\sin\theta \cdot \cos(\psi - \frac{\pi}{2}), \\ W_z &= \rho\sin\theta \cdot \sin(\psi - \frac{\pi}{2}). \end{aligned} \tag{5}$$

Also, direction vector δ_i, which is defined by $(u_i^k - v_i^k) / \| u_i^k - v_i^k \|$, is generated.

2.1.2 Line Clustering

After extraction of line segments, clustering is processed to group lines that belong to the same plane. If the

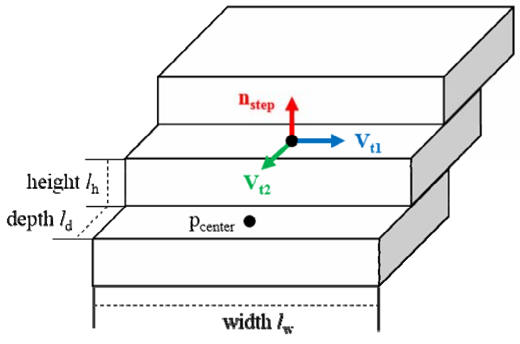

Fig. 3. Parameters of staircases.

points of a plane are obtained by our scanning system, the directions of the lines are similar. The line extracted in each scan is compared with representative direction δ_j using the following condition:

$$|\delta_j \cdot \delta_m| > \delta_c \tag{6}$$

where δ_m is the direction vector that represents each cluster. If Eq. (6) is satisfied, the line is included in the cluster, and δ_m is recalculated by simple moving average (SMA):

$$\delta_m' = (N_m \delta_m + \delta_j) / (N_m + 1) \tag{7}$$

where N_m represents the number of lines included in the cluster.

2.1.3 Plane Fitting

To parameterize the plane of each cluster, a line segment of cluster λ_i is selected randomly. And we select the two line segments, $\lambda_{i-1}, \lambda_{i+1}$, that are closest and second closest to λ_i. The plane parameters n_j and d_j are determined by calculating the singular value decomposition (SVD) on six end points of three selected line segments.

2.2 Large Plane Elimination

After large planes, such as walls, floors, and ceilings, are detected, a partial point cloud, $\mathbf{P}_{obj}$, is obtained by removing from $\mathbf{P}$ the points belonging to the large planes. Our proposed method detects staircases from $\mathbf{P}_{obj}$, and extracts several parameters, including dimensions (l_d, l_h, l_w) and poses (n_{step}) of the staircases, as shown in Fig. 3. This large-plane elimination process reduces the number of points significantly, and thus, improves efficiency of the proposed staircase detection method.

2.3 Staircase Detection

In the proposed method, the approximate positions of steps are extracted using supervoxel clustering. Then, the detected stairs are refined to optimize the parameters of the staircases. Finally, the staircases that have not been detected due to occlusion are predicted and generated virtually.

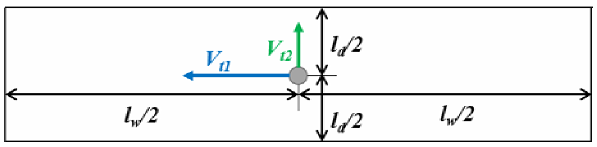

Fig. 4. Parameters of a step.

2.3.1 Supervoxel Clustering

In order to find the approximate position of a staircase, supervoxel clustering is employed. Supervoxel clustering groups points into perceptually meaningful regions that are homogeneous with respect to location and normal vectors. Supervoxel V_i is generated by using supervoxel clustering. Based on the fact that the plane of each step is parallel to the floor plane, we eliminate supervoxels where normal vector n_{vi} does not satisfy the following condition:

$$|n_{floor} \cdot n_{vi}| > \delta_n, \tag{8}$$

where n_{floor} represents the normal vector of the floor plane.

In order to get rid of supervoxels corresponding to other objects, rather than staircases, supervoxels having similar height with respect to the floor plane are merged into the same cluster. A cluster corresponding to a step usually consists of several supervoxels. Therefore, a cluster having a small number of supervoxels is eliminated. Points belonging to each cluster are denoted by $\mathbf{P}_{step}$.

2.3.2 Initial Staircase Extraction

To extract the parameters from each cluster, principal component analysis (PCA) is applied to $\mathbf{P}_{step}$. Then, eigenvectors represent two tangential vectors and one normal vector, as shown in Fig. 3.

To calculate l_w, we first calculate signed distances of all the points in $\mathbf{P}_{step}$ to a plane that has a normal vector of V_{t1} and that passes through the center of mass for $\mathbf{P}_{step}$, as shown in Fig. 4. Because only points of the staircase are projected, the maximum value of the projection distance is part of a width. So, l_w is calculated by addition of a projection distance using V_{t1} and V_{t2}. Calculation of l_d is similar to that of l_w.

2.3.3 Parameter Refinement

Unfortunately, the parameter values obtained in the previous process are incorrect, because $\mathbf{P}_{step}$ contains outliers as well. In order to refine the parameters of steps, a range for staircase R_1 is calculated using the l_w and l_d.

$$R_1 = \sqrt{l_w^2 + l_d^2} \tag{9}$$

Then, $\mathbf{P}_{refined}$ is obtained, which has points within the range of R_1 with respect to the center of mass of $\mathbf{P}_{step}$. In this process, points of other objects connected with the staircases, including fences and handrails, can be

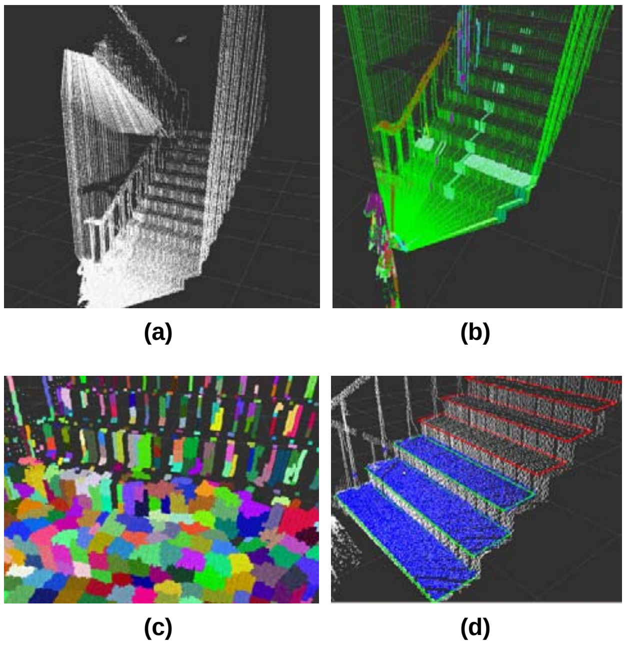

Fig. 5. The environment of stair detection: (a) The entire point cloud, (b) The plane extracted via line-based method [3], (c) The result of supervoxel clustering, (d) The staircases that are detected (green) and predicted (red).

eliminated. Next, $\mathbf{P}_{\text{refined}}$ is projected to n_{step} and extracts points $\mathbf{P}_{\text{step}}$, which can determine a step. To calculate the parameter of a step, we use the same process as in subsection 2.3.2.

2.3.4 Virtual Staircase Prediction

Due to occlusion and viewing angle, a small number of points are sensed for higher steps, and thus, a few steps are detected. Thereafter, we generate virtual steps to use the information effectively. Because each parameter of a step is similar, we can calculate predicted parameters using a mean of the calculated parameters.

3. Experimental Results and Conclusions

To implement our proposed method, we used a rotating 2D scanning LIDAR with an actuator in the environment, as shown in Fig. 5(a). At first, we applied a line segment–based plane detection method [3], as shown in Fig. 5(b). With the parameter values to extract planes of small sizes, the method cannot find large planes because of noise. In addition, an incorrect plane segmenting the steps vertically was also detected.

The result of supervoxel clustering is shown in Fig. 5(c). Compared with lower steps, supervoxels corresponding to upper steps do not have enough information to calculate parameters because of sparsity of points.

The green rectangles shown in Fig. 5(d) represent the staircase calculated by our proposed method. For the higher steps, we generated virtual steps represented by red rectangles using calculated parameter values.

We proposed a supervoxel clustering–based stair detection method to obtain the position and orientation of staircases. Before applying our method, we eliminate large planes, including walls, floors, and ceilings. Then, staircases are initially detected by using supervoxel clustering. Finally, unextracted staircases are predicted. As a result, staircases are calculated and predicted successfully.

References

[1] T. Tang, W. Lui and W. Li, "Plane-based detection of staircases using inverse depth", in Proc. Of ACRA 2012, Dec. 2012. Article (CrossRef Link)

[2] M. Y. Yang and W. Forstner, "Plane detection in point cloud data", Technischer Reprort, Institute of Geodesy and Geoinformation, Dept. of Phothgrammetry, 2010. Article (CrossRef Link)

[3] S. Y. An, L. K. Lee and S. Y. Oh, "Line segment-based fast 3D plane extraction using nodding 2D laser range finder", Robotica, pp. 1-24, September 2014. Article (CrossRef Link)

[4] G. A. Borges and J. J. Aldon, "Line extraction in 2D range images for mobile robotics", Journal of Intelligent and Robotics Systems, Vol. 40, pp. 267-297, Jul. 2004, doi:10.1023/B:JINT.0000038945.55712.65 Article (CrossRef Link)