

# Structure-From-Motion in 3D Space Using 2D Lidars

Dong-Geol Choi <sup>1</sup>, Yunsu Bok <sup>2,\*</sup>, Jun-Sik Kim <sup>3</sup>, Inwook Shim <sup>1</sup> and In So Kweon <sup>1</sup>

<sup>1</sup> Robotics and Computer Vision Lab, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea; dgchoi@rcv.kaist.ac.kr (D.-G.C.); iwshim@rcv.kaist.ac.kr (I.S.); iskweon@kaist.ac.kr (I.S.K.)

<sup>2</sup> Broadcasting Media Research Lab, Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea

<sup>3</sup> Center for Robotics Research, Korea Institute of Science and Technology, Seoul 02792, Korea; junsik.kim@kist.re.kr

\* Correspondence: ysbok@etri.re.kr; Tel.: +82-42-860-4970

Academic Editor: Angelo Maria Sabatini

Received: 27 September 2016; Accepted: 17 January 2017; Published: 3 February 2017

**Abstract:** This paper presents a novel structure-from-motion methodology using 2D lidars (Light Detection And Ranging). In 3D space, 2D lidars do not provide sufficient information for pose estimation. For this reason, additional sensors have been used along with the lidar measurement. In this paper, we use a sensor system that consists of only 2D lidars, without any additional sensors. We propose a new method of estimating both the 6D pose of the system and the surrounding 3D structures. We compute the pose of the system using line segments of scan data and their corresponding planes. After discarding the outliers, both the pose and the 3D structures are refined via nonlinear optimization. Experiments with both synthetic and real data show the accuracy and robustness of the proposed method.

**Keywords:** 2D lidar; structure-from-motion; pose estimation

## 1. Introduction

Two-dimensional lidars (Light Detection And Ranging) have been one of the most important sensors in many robotic applications because of its accuracy and robustness at measuring distance. It provides two-dimensional distance measurements on its own scanning plane. Many robotic systems [1–4] that require localization or even simultaneous localization and mapping (SLAM) have successfully utilized 2D lidars as their essential sensor. In recent years, many low-cost depth cameras have become popular (e.g., Microsoft Kinect, Intel RealSense, etc.) and have been used for various applications, such as photometric vision [5,6], object recognition [7,8] and odometry estimation [9–11]. However, 2D lidars are still preferred because of their long range and robustness to lighting conditions.

Compared to depth cameras, one of the key drawbacks of 2D lidars is that it provides only two-dimensional distance measurements at a time. It usually consists of a laser range finder and a rotating mirror to measure distances of scene points on a scan plane. Two-dimensional localization and mapping [12] is a popular application of 2D lidars, which is useful for wheeled robots and ground vehicles. It works based on the assumption that the motion of the lidar is two-dimensional, that the measurements have been collected at a consistent height. Once the assumption is broken due to the tilting motion of the lidar, there is no way to estimate the six degree-of-freedom (DOF) motion of the lidar.

Several approaches have been proposed to handle 2D lidars in a 3D space. The first approach is to fuse another motion information with the lidar measurements. Many systems utilize inertial measurement units (IMUs) to compensate for the tilting motion of the sensor [13–15]. However, this sensor fusion mostly aims to make the measurements of the tilted sensor match the 2D map, so that the compensated measurements can be used for 2D applications.



The second and most preferred solution is to build a 3D lidar system. One 2D lidar on a rotating stage provides 3D measurements around the sensor [16–18]. This 3D scanning device becomes an essential sensor for outdoor and automated systems such as self-driving cars [19–22]. Most of the successful systems participating in the defense advanced research projects agency (DARPA) urban challenge [23], which requires self-driving capability in urban environments, have been equipped with the 3D lidar system. Zhang and Singh [24] rotated a 2D lidar to obtain depth information. The advantage of their work is that the whole 3D structure is captured in a static pose, while the proposed system must move. To use their system, however, the motor must also be calibrated with high accuracy because their work assumes that the rotation speed is known. Moreover, their algorithm has a probability of divergence in the optimization process because it considers the system's motion during a single 3D scan.

The final approach is to use visual information along with the lidar measurements [25]. Because vision sensors provide abundant information about the scene, fusion of the visual information and the distance measurements by the 2D lidars enable 6D pose estimation. Pose estimation using cameras is related to 'structure-from-motion (SFM)', which is one of most popular issues in the area of computer vision. However, it requires many distinguishable visual features. Camera-based SFM fails if there are few visual features due to homogeneous areas.

In this paper, we propose a novel methodology of estimating the 6D pose of a system that consists of only 2D lidars. It is basically similar to the SFM framework, but is purely based on the measurements of 2D lidars. Instead of visual features corresponding to a 3D point cloud, we utilize the line segments of the scan data corresponding to the planar structures of a target scene. The pose of the system is initialized and refined using a number of the line–plane correspondences, and the 3D structures of the target scene are also updated via a nonlinear refinement process. The proposed algorithm is verified using real data. It successfully estimated the 3D structures and the 6D poses of the sensors without any additional sensors or motion constraint.

## 2. Overview of the Proposed Method

The flow chart of the proposed method is shown in Figure 1. It is divided into two steps, the initialization step and the SFM step. In the first step, we assume that the parameters of the three planes scanned by 2D lidars are known. Line-plane correspondences are designated manually, and the sensor pose is estimated using the correspondences (Section 3). The pose estimation is iterated until new planes are detected (Section 4.2). If we detect a new plane, we refine both the sensor poses and plane parameters by minimizing the squared sum of the Euclidean distances between the scan data and the corresponding plane (Section 4.3). The SFM process after the initialization step is similar to that of the initialization step, except the designation of line-plane correspondences is done automatically using the previous pose information (Section 4.1). After initializing the sensor pose, both the sensor poses and the plane parameters are refined via nonlinear optimization, even when a new plane is not detected.

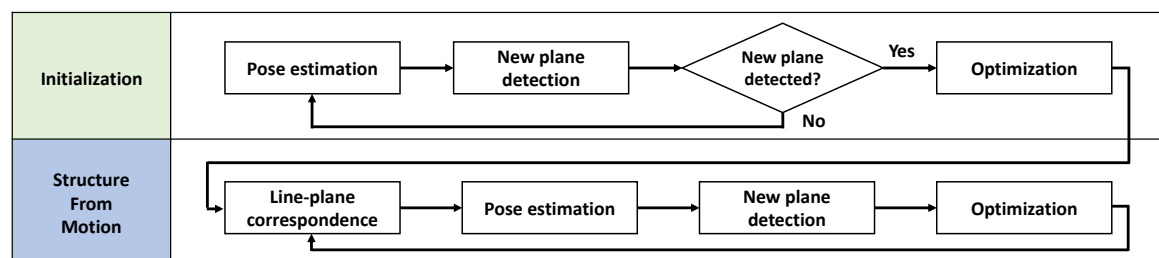


Figure 1. Flow chart of the proposed method.



### 3. Sensor Pose Estimation

This section presents the algorithm of estimating the pose of a sensor system that consists of 2D lidars. It is assumed that the system is fully calibrated. The sensor system and its own coordinate system are referred to as the ‘sensor’ and ‘sensor coordinate system’, respectively, for the rest of this paper.

#### 3.1. Pose Parameterization Using Two Lines on Two Planes

In this section, we show that the sensor pose is represented with two lines on two planes. Let us assume that we have a plane in the world coordinate system with known parameters,  $\Pi = [\mathbf{u}^\top d]^\top$ , where  $\mathbf{u}$  and  $d$  denote the unit normal vector and the constant term of the plane, respectively. If it is scanned by a 2D lidar, the intersecting line of the plane and the lidar’s scan plane appears in the scan data. The line can be defined by two points  $\mathbf{p}'$  and  $\mathbf{q}'$  in the sensor coordinate system. They are also represented as  $\mathbf{p} = \mathbf{R}\mathbf{p}' + \mathbf{t}$  and  $\mathbf{q} = \mathbf{R}\mathbf{q}' + \mathbf{t}$  in the world coordinate system, where  $[\mathbf{R} \ \mathbf{t}]$  is the sensor-to-world transformation, i.e., the sensor pose. The direction vector  $\mathbf{v} \equiv \mathbf{p} - \mathbf{q}$  of the line in the world coordinate system becomes simply

$$\mathbf{v} = \mathbf{R}\mathbf{v}', \quad (1)$$

where  $\mathbf{v}' \equiv \mathbf{p}' - \mathbf{q}'$  is the direction vector of the line in the sensor coordinate system.

The points  $\mathbf{p}$  and  $\mathbf{q}$  must lie on the plane  $\Pi$ , and this gives

$$\Pi^\top \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \mathbf{u}^\top (\mathbf{R}\mathbf{p}' + \mathbf{t}) + d = 0, \quad (2)$$

$$\mathbf{u}^\top \mathbf{R}(\mathbf{p}' - \mathbf{q}') = \mathbf{u}^\top \mathbf{R}\mathbf{v}' = 0. \quad (3)$$

Once we have multiple line–plane correspondences and the known rotation  $\mathbf{R}$ , the translation  $\mathbf{t}$  is simply estimated using Equation (2) via a linear estimation. Thus, we focus on the estimation of the rotation matrix  $\mathbf{R}$ .

We first show that the rotation  $\mathbf{R}$  can be decomposed into the two parts, one computed using the direction vectors in the world coordinate system and the other computed using the direction vectors in the sensor coordinate system.

Let us assume that we have two scan lines  $\mathbf{l}_1$  and  $\mathbf{l}_2$  corresponding to two planes  $\Pi_1$  and  $\Pi_2$ , respectively, as shown in Figure 2. The direction vector of  $\mathbf{l}_n$  is defined as  $\mathbf{v}_n$  in the world coordinate system, and  $\mathbf{v}'_n$  in the sensor coordinate system. It should be noted that the plane parameters  $\Pi_n$  are defined only in the world coordinate system.

We set an intermediate coordinate system, whose  $x$ - $z$  plane is equal to the plane spanned by  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , and its  $x$ -axis coincides with  $\mathbf{v}_1$ . The rotation  $\mathbf{R}_{iw}$  from the intermediate coordinate system to the world coordinate system becomes

$$\mathbf{R}_{iw} = \begin{bmatrix} \mathbf{r}_{iw}^1 & \mathbf{r}_{iw}^2 & (\mathbf{r}_{iw}^1 \times \mathbf{r}_{iw}^2) \end{bmatrix}, \quad (4)$$

where

$$\mathbf{r}_{iw}^1 = \frac{\mathbf{v}_1}{|\mathbf{v}_1|}, \quad \mathbf{r}_{iw}^2 = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{|\mathbf{v}_1 \times \mathbf{v}_2|}. \quad (5)$$

Similarly, the rotation  $\mathbf{R}_{is}$  from the intermediate coordinate to the sensor coordinate is

$$\mathbf{R}_{is} = \begin{bmatrix} \mathbf{r}_{is}^1 & \mathbf{r}_{is}^2 & (\mathbf{r}_{is}^1 \times \mathbf{r}_{is}^2) \end{bmatrix}, \quad (6)$$

where

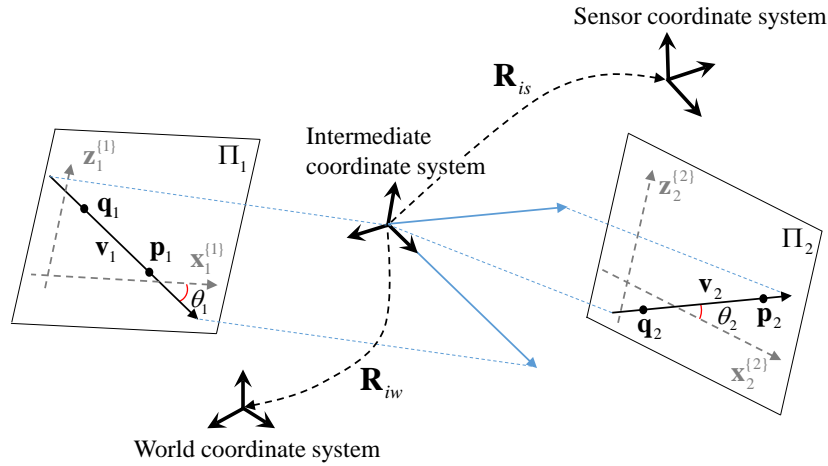
$$\mathbf{r}_{is}^1 = \frac{\mathbf{v}'_1}{|\mathbf{v}'_1|}, \quad \mathbf{r}_{is}^2 = \frac{\mathbf{v}'_1 \times \mathbf{v}'_2}{|\mathbf{v}'_1 \times \mathbf{v}'_2|}. \quad (7)$$



Thus, the rotation  $\mathbf{R}$  from the sensor coordinate system to the world coordinate system is simply

$$\mathbf{R} = \mathbf{R}_{iw} \mathbf{R}_{is}^\top. \quad (8)$$

Note that  $\mathbf{R}_{iw}$  is dependent only on  $\mathbf{v}_n$ , while  $\mathbf{R}_{is}$  is only dependent on  $\mathbf{v}'_n$ . From the sensor measurements,  $\mathbf{R}_{is}$  is already known as well as the direction vectors  $\mathbf{v}'_n$ . To estimate the rotation matrix completely, we need only two corresponding direction vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  in the world coordinate system.



**Figure 2.** Parameterization of rotation matrix using two line angles,  $\theta_1$  and  $\theta_2$ , on the scene planes. The sensor-to-world rotation matrix  $\mathbf{R}$  is expressed as two rotation matrices  $\mathbf{R}_{iw}$  and  $\mathbf{R}_{is}$ . The rotation matrix  $\mathbf{R}_{iw}$  is only dependent on  $\theta_1$  and  $\theta_2$ . From the sensor measurement,  $\mathbf{R}_{is}$  is already known as well as the direction vector  $\mathbf{v}'_1$  and  $\mathbf{v}'_2$ .

Now, we represent the direction vectors  $\mathbf{v}_n$  in terms of angles  $\theta_n$ , which will be referred to as ‘line angles’ for the rest of this paper. We define the coordinate system of each plane  $\Pi_n$  so that its  $y$ -axis coincides with the plane normal  $\mathbf{u}_n$ . Then, the direction vector in the plane coordinate system can be expressed as

$$\mathbf{v}_n^{\{n\}} = [\cos \theta_n \quad 0 \quad \sin \theta_n]^\top, \quad (9)$$

where  $\mathbf{v}_n^{\{k\}}$  denotes the direction vector  $\mathbf{v}_n$  in the  $\Pi_k$ -coordinate system. The direction vector  $\mathbf{v}_n$  in the world coordinate system is equal to  $\mathbf{v}_n = \mathbf{R}_{\Pi_n} \mathbf{v}_n^{\{n\}}$ , where  $\mathbf{R}_{\Pi_n}$  is the fixed rotation matrix from the  $\Pi_n$ -coordinate system to the world coordinate system. Until now, the rotation estimation from the sensor to the world coordinate systems are parameterized with two line angles, one for  $\mathbf{v}_1^{\{1\}}$  and the other for  $\mathbf{v}_2^{\{2\}}$  on two different planes.

One may wonder how to set up the plane coordinate system with just the plane normal. Of course, there is one-dimensional freedom to choose the  $x$ - and  $z$ -axis. It can be set arbitrarily. Because we compute the line angle on the plane coordinate system, different selections of coordinate axes induce a different line angle matching the coordinate selection. Thus, the estimated rotation is invariant to the selection of plane coordinate systems.

In the following section, we will derive how to estimate the line angle  $\theta_n$  in the scene plane.

### 3.2. Estimation of the Line Angles on the Planes

As mentioned in Section 3.1, the sensor-to-world transformation is obtained if two line angles  $\theta_1$  and  $\theta_2$  in each plane coordinate system are known. To estimate the angles, we use the invariant property of angle between direction vectors: the angle between two direction vectors in the sensor



coordinate system should be preserved in the world or plane coordinate system. The inner product  $\alpha_2$  between two direction vectors is simply measurable in the sensor coordinate system and is calculated in  $\Pi_2$ -coordinate system as

$$\alpha_3 = \mathbf{v}_2'^\top \mathbf{v}_1' = \mathbf{v}_2^{\{2\}\top} (\mathbf{A} \mathbf{v}_1^{\{1\}}), \quad (10)$$

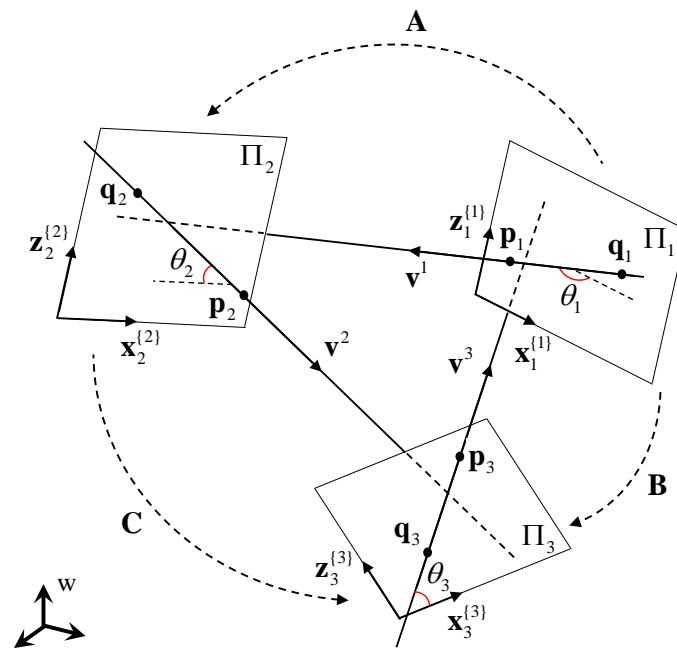
where  $\mathbf{A}$  is a rotation matrix from  $\Pi_1$ -coordinate to  $\Pi_2$ -coordinate, i.e.,  $\mathbf{A} = \mathbf{R}_{\Pi_2}^\top \mathbf{R}_{\Pi_1}$ , which is known.

Equation (10) contains two unknown variables  $\theta_1$  and  $\theta_2$  in  $\mathbf{v}_1^{\{1\}}$  and  $\mathbf{v}_2^{\{2\}}$ , respectively. Thus, it is not possible to solve it directly with only a single constraint Equation (10). This is resolved by introducing another line  $\mathbf{l}_3$  on plane  $\Pi_3$ . By adding one line, we have one more unknown  $\theta_3$  and two more constraint equations from the in-between angles as

$$\alpha_2 = \mathbf{v}_3'^\top \mathbf{v}_1' = \mathbf{v}_3^{\{3\}\top} (\mathbf{B} \mathbf{v}_1^{\{1\}}), \quad (11)$$

$$\alpha_1 = \mathbf{v}_3'^\top \mathbf{v}_2' = \mathbf{v}_3^{\{3\}\top} (\mathbf{C} \mathbf{v}_2^{\{2\}}), \quad (12)$$

where  $\mathbf{B} = \mathbf{R}_{\Pi_3}^\top \mathbf{R}_{\Pi_1}$  and  $\mathbf{C} = \mathbf{R}_{\Pi_3}^\top \mathbf{R}_{\Pi_2}$  are the rotation matrices from  $\Pi_1$ - to  $\Pi_3$ -coordinate system and from  $\Pi_2$ - to  $\Pi_3$ -coordinate system, respectively. Now, we have three trigonometric equations with three unknowns. The geometric relation is shown in Figure 3.



**Figure 3.** Three lines on three planes. From the three known angles among the lines, we drive three equations of  $\theta_1$ ,  $\theta_2$  and  $\theta_3$ . **A**, **B** and **C** indicate the known rotation matrices from  $\Pi_1$ - to  $\Pi_2$ -coordinate system, from  $\Pi_1$ - to  $\Pi_3$ -coordinate system, and from  $\Pi_2$ - to  $\Pi_3$ -coordinate system, respectively.

The elements of **A**, **B** and **C** at the  $i$ -th row and the  $j$ -th column are denoted by  $a_{ij}$ ,  $b_{ij}$  and  $c_{ij}$ , respectively. Equations (10) and (11) are expressed in terms of  $\theta_n$ :

$$k_1 c_2 + k_2 s_2 = \alpha_2, \quad (13)$$

$$k_3 c_3 + k_4 s_3 = \alpha_3, \quad (14)$$



where we defined that

$$c_n \equiv \cos \theta_n, \quad (15)$$

$$s_n \equiv \sin \theta_n, \quad (16)$$

$$k_1 \equiv a_{11}c_1 + a_{13}s_1, \quad (17)$$

$$k_2 \equiv a_{31}c_1 + a_{33}s_1, \quad (18)$$

$$k_3 \equiv b_{11}c_1 + b_{13}s_1, \quad (19)$$

$$k_4 \equiv b_{31}c_1 + b_{33}s_1. \quad (20)$$

From Equation (13) and  $c_2^2 + s_2^2 = 1$ , we obtain  $c_2$  and  $s_2$  as functions of  $\theta_1$ :

$$c_2 = \frac{\alpha_2 k_1 \pm k_2 \sqrt{k_1^2 + k_2^2 - \alpha_2^2}}{k_1^2 + k_2^2}, \quad (21)$$

$$s_2 = \frac{\alpha_2 k_2 \mp k_1 \sqrt{k_1^2 + k_2^2 - \alpha_2^2}}{k_1^2 + k_2^2}. \quad (22)$$

Similarly, we derive  $c_3$  and  $s_3$  as functions of  $\theta_1$  from Equation (14) and  $c_3^2 + s_3^2 = 1$ :

$$c_3 = \frac{\alpha_1 k_3 \pm' k_4 \sqrt{k_3^2 + k_4^2 - \alpha_1^2}}{k_3^2 + k_4^2}, \quad (23)$$

$$s_3 = \frac{\alpha_1 k_4 \mp' k_3 \sqrt{k_3^2 + k_4^2 - \alpha_1^2}}{k_3^2 + k_4^2}. \quad (24)$$

The sets of double signs,  $(\pm, \mp)$  and  $(\pm', \mp')$ , are independent. We substitute Equations (21)–(24) into Equation (12)

$$k_{11} - k_{12} = \mp \left( \pm' 2(k_7 k_8 + k_9 k_{10}) \sqrt{k_5 k_6} \right), \quad (25)$$

where we define that

$$k_5 \equiv k_1^2 + k_2^2 - \alpha_2^2, \quad (26)$$

$$k_6 \equiv k_3^2 + k_4^2 - \alpha_1^2, \quad (27)$$

$$k_7 \equiv \alpha_3 (c_{11}k_2k_3 - c_{13}k_1k_3 + c_{31}k_2k_4 - c_{33}k_1k_4), \quad (28)$$

$$k_8 \equiv \alpha_2 (c_{11}k_1k_4 + c_{13}k_2k_4 - c_{31}k_1k_3 - c_{33}k_2k_3), \quad (29)$$

$$k_9 \equiv \alpha_3 \alpha_2 (-c_{11}k_1k_3 - c_{13}k_2k_3 - c_{31}k_1k_4 - c_{33}k_2k_4) + \alpha_1 (k_1^2 + k_2^2) (k_3^2 + k_4^2), \quad (30)$$

$$k_{10} \equiv c_{11}k_2k_4 - c_{13}k_1k_4 - c_{31}k_2k_3 + c_{33}k_1k_3, \quad (31)$$

$$k_{11} \equiv k_5 k_7^2 + k_6 k_8^2, \quad (32)$$

$$k_{12} \equiv k_9^2 + k_5 k_6 k_{10}^2. \quad (33)$$

Squaring both sides of Equation (25) to remove the root sign, we obtain a 16th order equation with two variables,  $c_1$  and  $s_1$ . For convenience of calculation, we change the equation to a 16th order polynomial equation of a single variable. We multiply the squared Equation (25) by  $1/s_1^{16}$ , and every term of the equation can be expressed as

$$h \left( \frac{c_1}{s_1} \right)^{\sigma_1} \left( \frac{1}{s_1^2} \right)^{\sigma_2}, \quad (34)$$

$$(\sigma_1 \geq 0, \sigma_2 \geq 0, \sigma_1 + 2\sigma_2 = 16),$$



where we defined that the  $h$  is a coefficient value, and the  $\sigma_1$  and the  $\sigma_2$  are integer values. Substituting  $1 + (c_1^2/s_1^2)$  to  $1/s_1^2$ , the squared Equation (25) is expressed as a 16th order polynomial equation of  $c_1/s_1$ :

$$h \left( \frac{c_1}{s_1} \right)^{\sigma_1} \left( 1 + \frac{c_1^2}{s_1^2} \right)^{\sigma_2} = h \left( \frac{c_1}{s_1} \right)^{\sigma_1} + h \left( \frac{c_1}{s_1} \right)^{\sigma_1+2\sigma_2}. \quad (35)$$

From the 16th order polynomial equation, we obtain 16 candidates of  $c_1/s_1$ , which is equal to  $1/\tan\theta_1$ , within the range from  $-\pi/2$  to  $\pi/2$ . Because  $\tan(\theta_1 + \pi)$  is equal to  $\tan\theta_1$ , the number of candidates is doubled. They are substituted into Equations (21)–(24) to calculate  $\theta_2$  and  $\theta_3$ . Finally, we obtain combinations of the three angles,  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  that satisfy Equation (3) and Equation (12).

### 3.3. Physical Constraint for Solution Selection

From the solutions of the rotation matrix  $\mathbf{R}$  that were calculated in Section 3.2, we obtain solutions of the translation vector  $\mathbf{t}$  using a least square method with singular value decomposition (SVD)

$$\begin{bmatrix} \mathbf{u}_n^\top & \mathbf{u}_n^\top \mathbf{R} \mathbf{p}'_n + d_n \\ \mathbf{u}_n^\top & \mathbf{u}_n^\top \mathbf{R} \mathbf{q}'_n + d_n \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ 1 \end{bmatrix} = 0. \quad (36)$$

All the estimated poses  $\mathbf{R}$  and  $\mathbf{t}$  satisfy the given conditions Equation (3) and Equations (10)–(12), but there exist physically non-realizable poses. To choose only physically realizable poses, we adopt a new physical constraint that the location of the sensor must be in front of all planes.

A plane divides the 3D space into two sides, a front side and a back side. In our implementation, we set the normal directions of planes as their back side (the sign of  $d_n$  is also determined simultaneously). Since the sensor should be located only in front of the planes, the inner product of the plane normals and the sensor position vector must be negative:

$$\Pi_n^\top \mathbf{t} < 0. \quad (37)$$

Multiple pose candidates may meet this condition, and, in our experience, there are two physically realizable poses in most cases. It is impossible to distinguish which one is true. We can further choose the true six-dimensional sensor pose using an additional line–plane correspondence or an assumption that the pose of the sensor does not change much compared to the previous one, in the case of continuous sequences.

The whole process of the proposed method is described in Algorithm 1.

---

#### Algorithm 1 6-DOF Pose Estimation of a Lidar System using Three Lines

---

##### INPUT

- (a) Six points on the three scan lines (two in each line) in the sensor coordinate system
- (b) Parameters of three planes in the world coordinate system
- (c) One-to-one correspondences between scan lines and planes.

**OUTPUT:** Sensor-to-world transformation,  $\mathbf{R}$  and  $\mathbf{t}$

---

1. Compute  $\alpha_n$  using three line vector,  $\mathbf{v}_n^{\{n\}} = \mathbf{p}'_n - \mathbf{q}'_n$ , in the sensor coordinate system. ( $n = 1-3$ )
  2. Compute three rotation matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  using plane parameters in the world coordinate system.
  3. Compute the coefficients of  $k_1$  to  $k_{12}$ .
  4. Compute the coefficients of 16th order polynomial equation using  $k_1$  to  $k_{12}$ .
  5. Solve the equation to obtain candidates of  $\theta_1$
  6. Obtain combinations of  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  from Equations (21) to (24).
  7. Extract combinations of  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  that satisfy Equation (3) and Equation (12).
  8. Determine solutions of the sensor-to-world transformation,  $\mathbf{R}$  and  $\mathbf{t}$ , which meet the conditions mentioned in Section 3.3.
-



## 4. Structure from Motion

### 4.1. Line-Plane Correspondence

As we mentioned before, the line-plane correspondences in the initialization step are designated manually. After the initialization, the correspondences are designated automatically using the previous pose information. A new scan line obtained from the sensor system can be defined by two points,  $\mathbf{p}'$  and  $\mathbf{q}'$ , in the sensor coordinate system. We transform the points into the world coordinate system using the previous pose  $[\mathbf{R}_{pre} \ \mathbf{t}_{pre}]$ , which gives

$$\begin{aligned}\hat{\mathbf{p}} &= \mathbf{R}_{pre} \mathbf{p}' + \mathbf{t}_{pre}, \\ \hat{\mathbf{q}} &= \mathbf{R}_{pre} \mathbf{q}' + \mathbf{t}_{pre}.\end{aligned}\quad (38)$$

We compute the distance between two points and every plane and find the minimum value  $l_{min}$ :

$$l_{min} = \min_{i=1 \sim M} \sqrt{(\mathbf{u}_i^\top \hat{\mathbf{p}} + d_i)^2 + (\mathbf{u}_i^\top \hat{\mathbf{q}} + d_i)^2}, \quad (39)$$

where  $M$  is the number of planes. If  $l_{min}$  is smaller than a user-defined threshold, we designate the pair of the line and the plane as a line-plane correspondence. Lines with  $l_{min}$  bigger than the threshold do not correspond to any plane. They will be referred to as ‘non-assigned lines’ for the rest of this paper.

### 4.2. New Plane Detection

The non-assigned lines are accumulated as we repeat the structure-from-motion process. If the number of non-assigned lines is bigger than a user-defined number, we detect new planes using these lines. In this paper, the user-defined number was used as 30, which was determined empirically. We select two lines to generate a plane candidate and measure distances between the candidate and the remaining non-assigned lines. Those with the distance smaller than a user-defined threshold are classified as inliers of the plane candidate. Among the plane candidates generated by all combinations of non-assigned lines, we determine one with the largest number of inliers as a new plane.

### 4.3. Nonlinear Optimization

All variables in the whole sequence are refined via nonlinear optimization. The cost function  $f$  for the optimization is the squared sum of the Euclidean distances between the plane and lidar points in the world coordinate system

$$f(\Pi_1 \sim \Pi_M, \mathbf{R}_1 \sim \mathbf{R}_N, \mathbf{t}_1 \sim \mathbf{t}_N) = \sum_{i=1}^M \sum_{j \in A_i} \sum_{\mathbf{p} \in B_{i,j}} \Pi_i^\top (\mathbf{R}_j \mathbf{p} + \mathbf{t}_j), \quad (40)$$

where  $N$  is the number of scans of sensor data.  $A_i$  is the set of pose indices that scan the plane  $\Pi_i$ , and  $B_{i,j}$  is the set of points that lie on the plane  $\Pi_i$  and belong to the  $j$ -th pose. We adopt the Levenberg-Marquardt [26] algorithm for the nonlinear optimization.

The refinement may be achieved by conventional registration methods such as iterative closest point (ICP) [27,28]. However, the nonlinear optimization is better than the registration because of three reasons listed below:

- the correspondence between points and planes are not changed in the optimization process;
- the overlap between a single scan and a point cloud is very narrow (several lines on two scanning planes);
- the nonlinear optimization utilizes Jacobian while the registration does not.

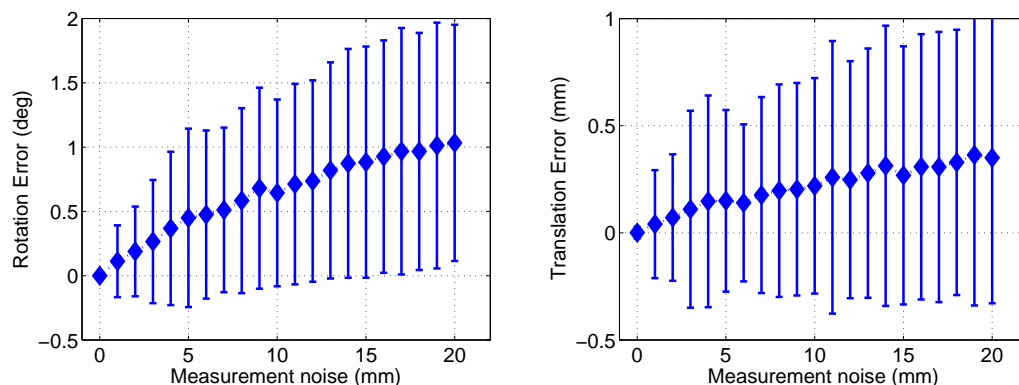


## 5. Experimental Results

We evaluated the performance of the proposed pose estimation and SFM algorithms through several experiments. In every experiment, the error of an estimated pose  $\mathbf{T}$  is defined as a residual transformation  $\delta\mathbf{T} = \mathbf{T}_{ref}^{-1}\mathbf{T}$  between the reference transformation  $\mathbf{T}_{ref}$  and the estimated pose. Rotation error is measured by the rotation angle of Rodrigues' rotation formula [29] of the rotation in the residual transformation, and the translation error is measured by the translation part of  $\delta\mathbf{T}$ .

### 5.1. Evaluation of Pose Estimation Algorithm

We performed two experiments that use synthetic and real data to evaluate the performance of the proposed pose estimation algorithm. For the synthetic experiment, we generate a set of data that consists of a sensor-to-world transformation, three plane parameters and six points on the planes. The transformation is generated randomly with its maximum rotation angle as 50 degrees and maximum translation as four meters along every axis. The plane parameters are also generated randomly with a maximum distance of five meters from the world origin. The points are generated in the interval of  $[-4, 4]$  meters along every axis from the world origin. Two of the three coordinates are generated randomly while the other is computed using the plane parameters to generate points 'on the planes'. We added white noise with varying intervals to the points and compared the result by the proposed algorithm to the ground truth of the sensor-to-world transformation. We performed 1000 trials and computed the mean and standard deviation of the rotation and translation error, shown in Figure 4. The measurement noise in the horizontal axis indicates the interval in which the noise is generated. Both the rotation error and translation error increase as the measurement noise increases. Although the proposed algorithm does not utilize many points to reduce the effect of noise, the result is not very sensitive to the noise of scan data. In real-world cases, moreover, techniques such as line fitting may reduce the effect of noise.



**Figure 4.** Experimental results using synthetic data. The value  $x$  of 'Measurement noise' (horizontal axis) means that the noise added to the points are generated in the interval of  $[-x, x]$ .

For the real data experiment, we designed a hand-held sensor system that consists of two 2D lidars (UTM-30LX, Hokuyo, Osaka, Japan) and three cameras (Flea3, PointGery, Richmond, BC, Canada,  $640 \times 480$  pixels). The lidars are pointed at the ceiling, and the two cameras are pointed at the side of the lidars (see Figure 5), and one camera is pointed at the forward direction of the system. The cameras are only used to acquire color information. The maximum scan rate of the system is about 20 Hz. The system is fully calibrated: camera intrinsic parameters using [30], lidar to camera [31,32] and between two lidars [33].





**Figure 5.** Hand-held sensor system for experimental validations. It consists of two 2D lidars headed to the ceiling and three cameras headed to the side of the lidars and heading to the direction of the system. We use only two 2D lidars for pose estimation. The cameras are attached to obtain color information. The maximum scan rate of the system is about 20 Hz. The system is fully calibrated, including both intrinsic and extrinsic parameters.

We captured three non-parallel checkerboard patterns in three different configurations (see Figure 6) to test the performance of the proposed method in various scene structures. For each configuration, the relative poses among the planes are computed using a number of images captured by two cameras with known intrinsic parameters. The ground plane is scanned by both lidars, while the other two are captured by each lidar and camera. The images captured by two cameras are used to compute the ‘reference’ relative poses between the planes and the sensor system. Among all the scans, we selected 351 scans whose images do not include serious motion blur (the sensor was in motion while the data is captured). The projection errors of the checkerboard corners in the selected scans are smaller than one pixel.



**Figure 6.** Three non-parallel planes in three different configurations are captured by the sensor system in Figure 5.

The distribution of the difference between the poses computed using laser scans (proposed) and checkerboard images (reference) are displayed in Figure 7, which is estimated by the kernel density estimation method. We show the error distributions of the three different configurations. Most scans have rotation errors smaller than four degrees and translation errors smaller than four millimeters. The mean and standard deviation values of the pose errors are shown in Table 1. The experimental results shows that the proposed method performs robustly in various plane configurations.



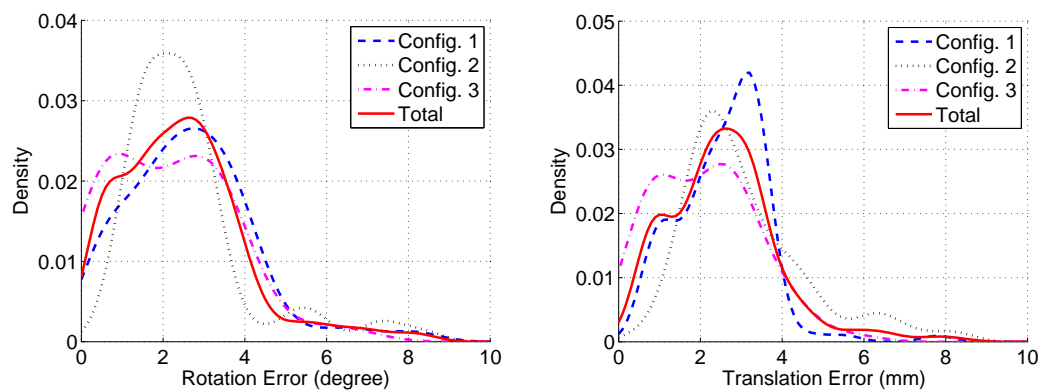


Figure 7. Distributions of the pose difference between the estimated and the reference poses.

Table 1. The pose errors in different plane configurations.

Configuration	Number of Scans	Rotation Error	Translation Error
		Mean(std) in Degree	Mean(std) in mm
1	113	2.6431 (1.5471)	2.5639 (1.0834)
2	108	2.6104 (1.5523)	3.0944 (1.5391)
3	130	2.1617 (1.4839)	2.1124 (1.2203)
Total	351	2.4548 (1.5379)	2.5599 (1.3458)

## 5.2. Evaluation of Structure-from-Motion Algorithm

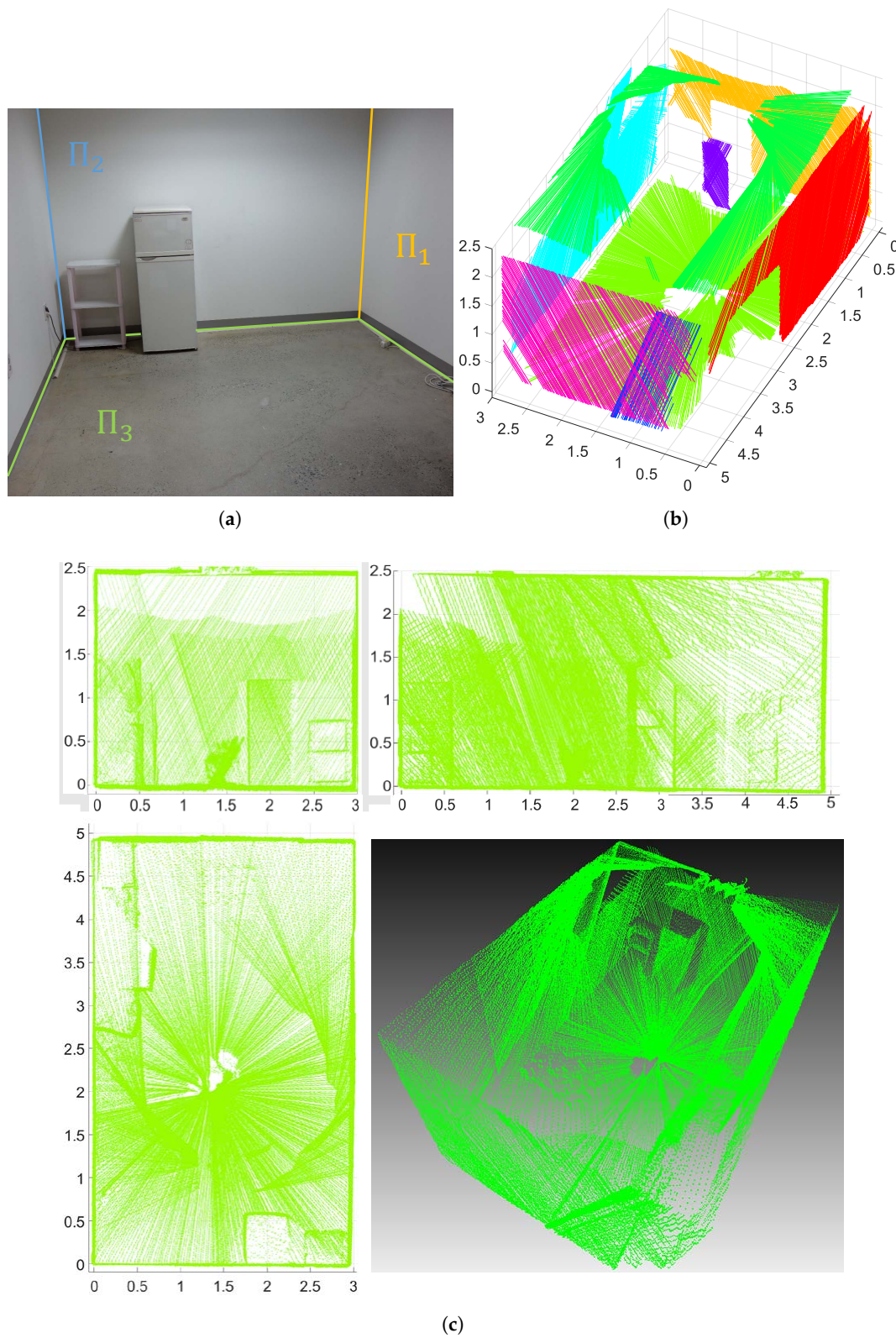
The proposed SFM algorithm can be used in many applications, especially localization and mapping. All processes except data collection (i.e., motion estimation, structure-from-motion, and nonlinear-optimization) were performed offline. We set two geometric parameters for the experiments. Only lines longer than 0.5 m were used for line extraction with lidar's scan points, and the inlier bound of the plane was set at 0.05 m. These parameters are determined empirically. For the first experiment, we scanned a small room using the sensor system continuously and estimated the pose of each scan. As shown in Figure 8a, three perpendicular planes,  $\Pi_1$ ,  $\Pi_2$  and  $\Pi_3$  are used as references for the initialization step. It should be noted that we extracted four lines from two lidar scans—lines corresponding to  $\Pi_1$  and  $\Pi_3$  from one lidar, and lines corresponding to  $\Pi_2$  and  $\Pi_3$  from the other lidar—to determine a unique solution for each scan, as mentioned in Section 3.3. In Figure 8b, scanned line segments are displayed in colors of corresponding planes. The results shown in Figure 8c verifies that the proposed method gives accurate pose information of the system. Six-hundred-and-fifty scans are accumulated and eight planes are detected based on the poses to reconstruct the 3D structure of the room.

The size of the room measured from the reconstruction result is given in Table 2, and compared to that measured by a laser distance meter (GLM 80, Bosch, Stuttgart, Germany).

Table 2. The results of the room measurement.

(Unit: mm)	Width	Length	Height
Laser distance meter	2973.1	4918.7	2374.7
Proposed	2957.6	4924.1	2371.65
Error	15.5 (0.521%)	5.4 (0.110%)	3.05 (0.128%)

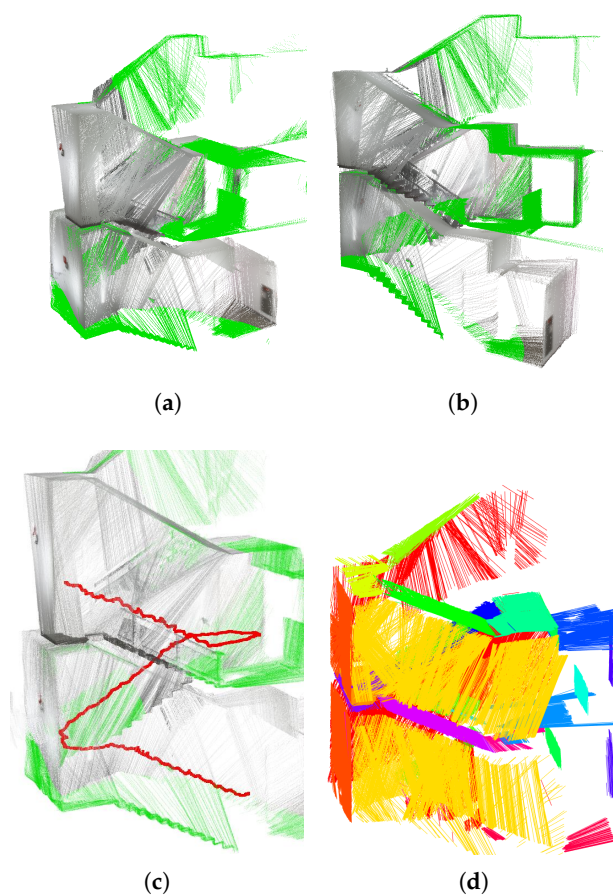




**Figure 8.** 3D reconstruction results of the proposed method: (a) real environment (b) scanned line segments displayed in colors of corresponding planes; and (c) 3D reconstruction results by accumulating scan data.

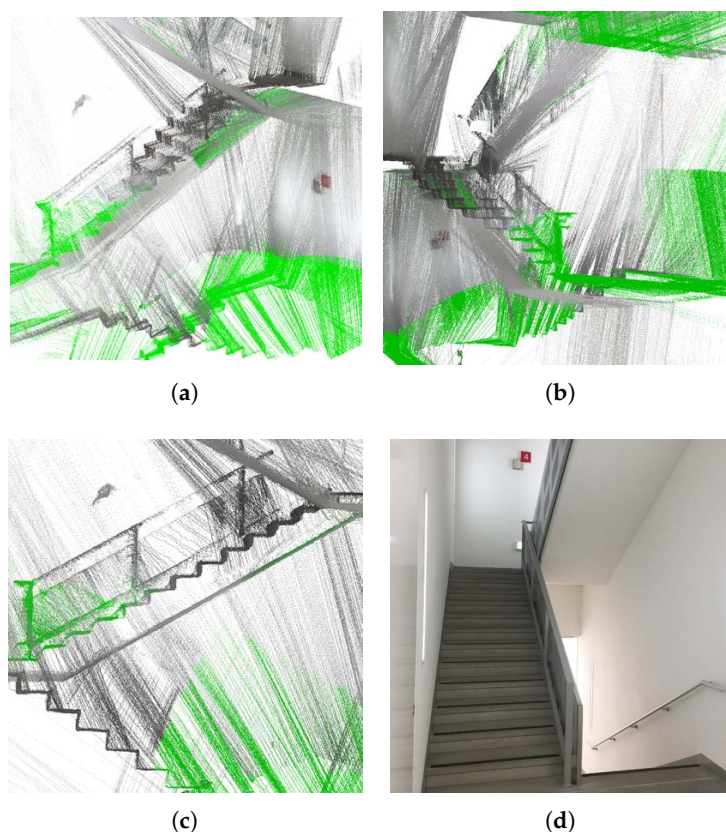


For the second experiment, we captured scan data while we passed through a stairway. Figure 9 shows the 3D reconstruction result of the target scene. As shown in Figure 10d, the environment is very challenging for vision based methods because it does not contain enough visual features. However, the proposed method does not suffer from the homogeneous areas because it requires only structural information, not visual features. Thus, it gives an accurate 3D reconstruction result as shown in Figure 9a,b. In Figure 9c, the locations of the sensor system (red dots) are overlaid on the 3D reconstruction result. Scanned line segments are also displayed in colors of corresponding planes, in Figure 9d. In this experiment, we detected 23 planes in total. The steps are not detected as planes because the stairs are scanned vertically – the scanning plane was perpendicular to the longer side of the stairs so that lines on them were shorter than the threshold (0.5 meters in our experiment). In Figure 10, we obtained color information of scan data from the images just for visualization. As shown in Figure 10a–c, we can recognize the shapes of both the handrail and the stairs. In this experiment, the sensor system moved about 25 m (down the stairway) in 125 s while it captured 3500 frames of scan data. This demonstrates the potential of the proposed method in lidar-only mapping and navigation.



**Figure 9.** Reconstruction results: (a,b) 3D reconstruction results using the proposed method; (c) estimated trajectory (red dots) of the sensor system with 3500 scans down the stairs; and (d) scanned line segments displayed in colors of corresponding planes.





**Figure 10.** Reconstruction results: (a–c) detailed views of the result with color information; (d) real environment.

## 6. Conclusions

In this paper, we have presented a novel structure-from-motion methodology using 2D lidars. The proposed algorithm uses only 2D range data, without any additional sensors. We have proposed a pose estimation method using three line-plane correspondences. To estimate the line angles, we have used the angles between measured lines and derived a 16th polynomial equation. We have also proposed a new structure-from-motion process using 2D lidar data. After discarding outliers, both the pose and the 3D structures were refined via nonlinear optimization. The experiments using real data validate that the proposed algorithm provides accurate and robust results in real environments.

There are several works that could improve the proposed system and method. The proposed system uses two 2D lidars to obtain three lines that lie on each different plane using a minimal number of sensors. We will try several sensor configurations to find the optimal line information. In this paper, we performed 3D reconstructions using the proposed method, but the result can be more accurate using recent loop closure techniques on scene matching. Additional sensors, such as GPS, and IMU, or even cameras, can be attached to improve the accuracy of the motion estimation.

**Acknowledgments:** This work was supported by the Technology Innovation Program (No. 2017-10069072) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea).

**Author Contributions:** D.-G. Choi designed and developed algorithms for the pose estimation and the structure-from-motion; Y. Bok designed the overall methodology and experiments; J.-S. Kim conceived and designed the pose estimation algorithm. I. Shim contributed implementation and experiment of the structure-from-motion algorithm; I.S. Kweon discussed the weaknesses of the system while it was being implemented and tested.

**Conflicts of Interest:** The authors declare no conflict of interest.



## References

1. Newman, P.; Cole, D.; Ho, K. Outdoor SLAM using visual appearance and laser ranging. In Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 1180–1187.
2. Nüchter, A.; Lingemann, K.; Hertzberg, J.; Surmann, H. 6D SLAM with approximate data association. In Proceedings of the 12th International Conference on Advanced Robotics, Seattle, WA, USA, 17–20 July 2005; pp. 242–249.
3. Cole, D.M.; Newman, P.M. Using laser range data for 3D SLAM in outdoor environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 1556–1563.
4. Choi, D.G.; Shim, I.; Bok, Y.; Oh, T.H.; Kweon, I.S. Autonomous homing based on laser-camera fusion system. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 2512–2518.
5. Han, Y.; Lee, J.Y.; Kweon, I.S. High quality shape from a single rgb-d image under uncalibrated natural illumination. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1617–1624.
6. Yu, L.F.; Yeung, S.K.; Tai, Y.W.; Lin, S. Shading-based shape refinement of rgb-d images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1415–1422.
7. Lai, K.; Bo, L.; Ren, X.; Fox, D. RGB-D object recognition: Features, algorithms, and a large scale benchmark. In *Consumer Depth Cameras for Computer Vision*; Springer: London, UK, 2013; pp. 167–192.
8. Schwarz, M.; Schulz, H.; Behnke, S. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 1329–1335.
9. Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 2100–2106.
10. Kerl, C.; Sturm, J.; Cremers, D. Robust odometry estimation for RGB-D cameras. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3748–3754.
11. Whelan, T.; Johannsson, H.; Kaess, M.; Leonard, J.J.; McDonald, J. Robust real-time visual odometry for dense RGB-D mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5724–5731.
12. Wolf, D.F.; Sukhatme, G.S. Mobile robot simultaneous localization and mapping in dynamic environments. *Auton. Robots* **2005**, *19*, 53–65.
13. Buckley, S.; Vallet, J.; Braathen, A.; Wheeler, W. Oblique helicopter-based laser scanning for digital terrain modelling and visualisation of geological outcrops. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 1–6.
14. Hesch, J.A.; Mirzaei, F.M.; Mariottini, G.L.; Roumeliotis, S.I. A Laser-aided Inertial Navigation System (L-INS) for human localization in unknown indoor environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010; pp. 5376–5382.
15. Bosse, M.; Zlot, R.; Flick, P. Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Trans. Robot.* **2012**, *28*, 1104–1119.
16. Patz, B.J.; Papeis, Y.; Pillat, R.; Stein, G.; Harper, D. A practical approach to robotic design for the darpa urban challenge. *J. Field Robot.* **2008**, *25*, 528–566.
17. Bosse, M.; Zlot, R. Continuous 3D scan-matching with a spinning 2D laser. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 4312–4319.
18. Sheehan, M.; Harrison, A.; Newman, P. Automatic self-calibration of a full field-of-view 3D n-laser scanner. In *Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 165–178.
19. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The stanford entry in the urban challenge. *J. Field Robot.* **2008**, *25*, 569–597.



20. Leonard, J.; How, J.; Teller, S.; Berger, M.; Campbell, S.; Fiore, G.; Fletcher, L.; Frazzoli, E.; Huang, A.; Karaman, S.; et al. A perception-driven autonomous urban vehicle. *J. Field Robot.* **2008**, *25*, 727–774.
21. Markoff, J. Google cars drive themselves, in traffic. *The New York Times*, 9 October 2010.
22. Shim, I.; Choi, J.; Shin, S.; Oh, T.H.; Lee, U.; Ahn, B.; Choi, D.G.; Shim, D.H.; Kweon, I.S. An autonomous driving system for unknown environments using a unified map. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1999–2013.
23. Buehler, M.; Iagnemma, K.; Singh, S. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 56.
24. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems Conference (RSS), Berkeley, CA, USA, 12–16 July 2014; pp. 109–111.
25. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181.
26. Marquardt, D.W. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, 431–441.
27. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. *Robotics-DL tentative*. **1992**, 586–606.
28. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155.
29. Belongie, S. Rodrigues' Rotation Formula. From MathWorld—A Wolfram Web Resource, Created by Eric W. Weisstein. 1999. Available online: <http://mathworld.wolfram.com/RodriguesRotationFormula.html> (accessed on 15 June 2003).
30. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334.
31. Zhang, Q.; Pless, R. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2301–2306.
32. Bok, Y.; Choi, D.G.; Kweon, I.S. Extrinsic calibration of a camera and a 2D laser without overlap. *Robot. Auton. Syst.* **2016**, *78*, 17–28.
33. Choi, D.G.; Bok, Y.; Kim, J.S.; Kweon, I.S. Extrinsic calibration of 2-D lidars using two orthogonal planes. *IEEE Trans. Robot.* **2016**, *32*, 83–98.



© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).