

Article

3D LIDAR-Camera Extrinsic Calibration Using an Arbitrary Trihedron

Xiaojin Gong *, Ying Lin and Jilin Liu

Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China; E-Mails: ly86@zju.edu.cn (L.Y.); liujl@zju.edu.cn (J.L.)

* Author to whom correspondence should be addressed; E-Mail: gongxj@zju.edu.cn; Tel.: +86-571-8795-1529.

Received: 14 November 2012; in revised form: 8 January 2013 / Accepted: 24 January 2013 / Published: 1 February 2013

Abstract: This paper presents a novel way to address the extrinsic calibration problem for a system composed of a 3D LIDAR and a camera. The relative transformation between the two sensors is calibrated via a nonlinear least squares (NLS) problem, which is formulated in terms of the geometric constraints associated with a trihedral object. Precise initial estimates of NLS are obtained by dividing it into two sub-problems that are solved individually. With the precise initializations, the calibration parameters are further refined by iteratively optimizing the NLS problem. The algorithm is validated on both simulated and real data, as well as a 3D reconstruction application. Moreover, since the trihedral target used for calibration can be either orthogonal or not, it is very often present in structured environments, making the calibration convenient.

Keywords: extrinsic calibration; 3D LIDAR-camera system; sensor fusion

1. Introduction

Multi-sensors are commonly equipped on mobile robots for navigation tasks. Currently, for instance, ranging sensors such as high-speed 3D LIDARs are often used in conjunction with cameras for a robot to detect objects [1,2] and reconstruct scenes [3–5]. In these sensor fusion-based applications, a prerequisite is to extrinsically calibrate the relative transformation between the sensors. The result of extrinsic calibration highly impacts subsequent fusion processes.

A variety of methods have been developed to address the LIDAR-camera extrinsic calibration problem. Among them, early interest focuses on systems consisting of a 2D LIDAR and a camera [6–9]. Wasielewski and Strauss [6] and Naroditsky *et al.* [7] calibrate a 2D laser scanner with respect to a camera by making use of special calibration rigs, such as a white planar board covered with a black line. The work of Zhang and Pless [8] relies on a planar checkerboard pattern. Corners of the pattern [10] are first detected in images and used to determine the poses of planes in camera frames. Meanwhile, 3D points falling on the checkerboard are taken into consideration to estimate the planes' poses in LIDAR frames. Using the geometric constraint of the planar target in a couple of LIDAR-camera observations, the extrinsic calibration problem is formulated as a nonlinear least squares (NLS) problem [11] and solved iteratively.

In recent years, with the development of 3D laser ranging techniques, several methods were proposed to calibrate 3D LIDAR-camera systems [12–18]. Unnikrishnan [12] and Pandey *et al.* [13] extend the checkerboard pattern-based method [8] from 2D to 3D LIDARs. Mirzaei *et al.* [19] utilizes a planar board covered with fiducial markers for calibration, which, in essence, is of the same rationale as the checkerboard-based approaches. They further divide the NLS optimization problem into two least-square sub-problems and solve them analytically. The checkerboard pattern is also used in the work of Geiger *et al.* [20]. They calibrate a 3D LIDAR-camera system using a single shot containing such multiple patterns. Instead of using planar checkerboard patterns, there are several alternative methods that rely on correspondences of points [21], lines [22] or circles [14], or employ inertial sensors [23,24]. Compared to plane-based approaches, most of these methods need to build point- or line-wise correspondences between images and LIDAR points. However, due to the lower and non-uniform resolution of LIDAR measurements, it is difficult to achieve high accuracy.

In this work, we propose a novel way to conduct the extrinsic calibration between a 3D LIDAR and a camera. In contrast to most of the published techniques, our method distinguishes itself in two aspects:

1. It takes advantage of a trihedron—which may or may not be orthogonal—for calibration. Such trihedral targets are ubiquitous in both indoor and outdoor structured environments, such as two adjacent walls of a building together with the floor. Hence, it is quite convenient for a robot to collect data for calibration. Compared to the aforementioned calibration rigs, the trihedral configuration is less likely to be perturbed even under severe weather conditions, and is easier to be captured.
2. In contrast to these calibration-rig-based methods that require a user to specify both the region of a plane in 3D LIDAR and the corners in images, our method requires fewer user inputs. Only the region of each plane of the trihedron in the sensors' data is needed. Moreover, the precision of the manual inputs does not make much of a difference.

To present the proposed method, we organize the remainder of this paper as follows. In Section 2, we first describe the extrinsic calibration problem via taking advantage of an trihedral calibration rig, and introduce the associated geometric and motion constraints. Section 3 presents the entire calibration procedure. Experiments conducted on both simulations and real data are exhibited in Section 4, followed with conclusions in Section 5.

2. Problem Description

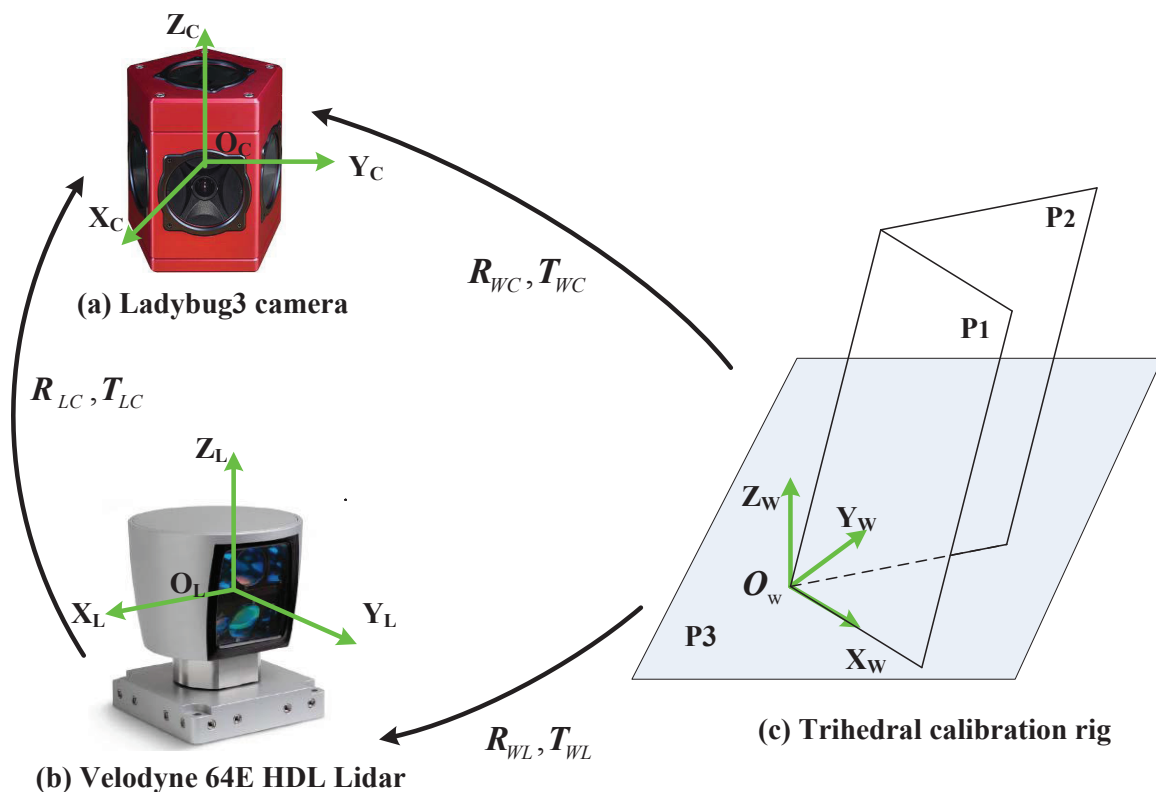
Let us formally define the problem of 3D LIDAR-camera extrinsic calibration. We are given a camera and a 3D LIDAR that are rigidly mounted with respect to each other. Both sensors are assumed to be pre-calibrated, meaning that their intrinsic parameters are known. A trihedron is observed synchronously by them. Our objective is to determine the relative transformation between the two sensors, by taking advantage of the constraints associated with the trihedron.

For the sake of clarity, in the remaining of this section, we introduce the related definitions and notations, together with the geometric and motion constraints established between the measurements of the two sensors.

2.1. Definitions and Notations

Figure 1 demonstrates a typical calibration configuration. It includes a system composed of a Ladybug3 omnidirectional camera [25] and a commercially available high-speed Velodyne HDL-64E 3D LIDAR [26], as well as a trihedral target viewed by both sensors. In experiments, the trihedron is fixed and the sensor system moves to obtain multiple configurations. In such configurations, several reference frames are defined:

Figure 1. A typical calibration configuration. (a) is a Ladybug3 camera and (b) is a Velodyne HDL-64E LIDAR. Both are rigidly assembled with respect to each other. A trihedral object (c), which may or may not be orthogonal, is observed by both sensors.



- **Camera frame:** The proposed method is not restricted to a specific camera type, as long as the camera is of a single viewpoint [27] and pre-calibrated. The camera frame is set up to be

coincident with the one defined in its projection model. The frame is represented by $\{C_i\}$, in which $i = 1 \dots N$ indicates the i^{th} configuration.

- **LIDAR frame:** The LIDAR frame is also defined to be coincident with the one in its own projection model, and is denoted by $\{L_i\}$.
- **World reference frame:** The world reference frame is fixed on the trihedron. Since the trihedral object can be either orthogonal or not, the reference frame is set up in such a way that the origin is at the common vertex and the axis X aligns with one intersection line. The axis Z is aligned with the direction of the plane P_3 's normal vector, and Y is further determined following the right-hand rule, as illustrated in Figure 1(c). The world frame is denoted by $\{W\}$.

Once the frames are defined, we represent the relative rotation and translation from one frame A to another frame B by \mathbf{R}_{AB} and \mathbf{T}_{AB} , where $A, B \in \{C_i, L_i, W\}$. Then, given a 3D point \mathbf{P}_A in the frame A , the corresponding point \mathbf{P}_B in B is computed via $\mathbf{P}_B = \mathbf{R}_{AB}\mathbf{P}_A + \mathbf{T}_{AB}$. In practice, the sensors are rigidly mounted on a mobile robot, and the transformations from the LIDAR to the camera, *i.e.* $\mathbf{R}_{L_iC_i}$ and $\mathbf{T}_{L_iC_i}$, are fixed in all configurations even when the robot moves. Hence, they are simply denoted by \mathbf{R}_{LC} and \mathbf{T}_{LC} , which are the parameters we aim to estimate in the calibration task.

In addition, we know that a plane in a frame is specified by $\mathbf{N}^T\mathbf{P} - d = 0$, where \mathbf{P} is an arbitrary 3D point lying on the plane, and \mathbf{N} and d are, respectively, the normal vector and the distance. Hence, we use $\{\mathbf{N}_A^j, d_A^j\}$ to describe the j^{th} plane of the trihedron with respect to the frame A , $j = 1 \dots 3$, and $\mathbf{P}_A^{j,k}$ specifies the k^{th} point on the plane.

2.2. Geometric and Motion Constraints

The proposed method makes use of a trihedron as a calibration rig. Hence, in order to address the extrinsic calibration problem, several constraints are taken into consideration. They are summarized as follows.

- **Trihedral constraint:** Let us consider the trihedron with respect to a sensor frame A . If its three planes, $\{\mathbf{N}_A^j, d_A^j\}$, are estimated, then the relative rotation \mathbf{R}_{WA} and translation \mathbf{T}_{WA} from the world frame to the sensor frame are uniquely determined. We represent $\mathbf{R}_{WA} = [\mathbf{r}_{WA1} \ \mathbf{r}_{WA2} \ \mathbf{r}_{WA3}]$, where \mathbf{r}_{WA1} , \mathbf{r}_{WA2} , and \mathbf{r}_{WA3} are column vectors. Then, we have

$$\begin{cases} \mathbf{r}_{WA3} = \mathbf{N}_A^3 \\ \mathbf{r}_{WA1} = \frac{\mathbf{N}_A^1 \times \mathbf{N}_A^3}{\|\mathbf{N}_A^1 \times \mathbf{N}_A^3\|} \\ \mathbf{r}_{WA2} = \mathbf{r}_{WA3} \times \mathbf{r}_{WA1} \end{cases} \quad (1)$$

and the translation is

$$\mathbf{T}_{WA} = \begin{bmatrix} \mathbf{N}_A^1 & \mathbf{N}_A^2 & \mathbf{N}_A^3 \end{bmatrix}^{-T} \begin{bmatrix} d_A^1 & d_A^2 & d_A^3 \end{bmatrix}^T \quad (2)$$

- **Planarity constraint between two frames:** This constraint implies that, if points in a frame A are coplanar, then they must lie on a plane when transformed to another frame B . It means that, in the absence of noise, we have a plane $\{\mathbf{N}_B, d_B\}$ such that

$$\sum_k \|\mathbf{N}_B^T(\mathbf{R}_{AB}\mathbf{P}_A^k + \mathbf{T}_{AB}) - d_B\|^2 = 0 \quad (3)$$

for all coplanar points \mathbf{P}_A^k in the frame A .

- **Planarity constraint between two images:** This constraint describes the relationship between a set of coplanar feature points and their correspondences in two images. Given a single-viewpoint camera, for the purpose of generality, we represent its projection model as $\mathbf{p} = \mathbf{F}(\mathbf{P})$, where \mathbf{P} is a 3D point in space and \mathbf{p} is the projected image point. The inverse projection model is specified by $\mathbf{P} = \gamma\mathbf{F}^{-1}(\mathbf{p})$, in which γ is an unknown scalar, meaning that \mathbf{P} lies on a ray determined by \mathbf{p} , but its distance stays unknown. Now, we consider two camera frames C_1 and C_2 . In the first frame, the plane on which all the features lie is defined by $\{\mathbf{N}_{C_1}, d_{C_1}\}$. Then, pair-wise corresponding image features \mathbf{p}_{C_1} and \mathbf{p}_{C_2} satisfy

$$\mathbf{p}_{C_2} - \mathbf{F} \left(\mathbf{R}_{C_1C_2} \frac{d_{C_1}\mathbf{F}^{-1}(\mathbf{p}_{C_1})}{\mathbf{N}_{C_1}^T\mathbf{F}^{-1}(\mathbf{p}_{C_1})} + \mathbf{T}_{C_1C_2} \right) = 0 \quad (4)$$

Note that this constraint is also known as the homography constraint [28] when \mathbf{F} is a pinhole camera projection model.

- **Motion constraint:** When a robot platform moves from one location to another, the translation of the camera and that of the LIDAR are equal to each other, as the sensors are fixed rigidly. Hence, we have

$$\mathbf{T}_{WC_1} - \mathbf{T}_{WC_2} = \mathbf{R}_{WC}\mathbf{R}_{WL}^{-1}(\mathbf{T}_{WL_1} - \mathbf{T}_{WL_2}) \quad (5)$$

3. Algorithm Description

In order to estimate the relative transformation between a 3D LIDAR and a camera, we capture N ($N \geq 2$) observations of a trihedron by both sensors. The sensors are individually calibrated in each configuration first to get their extrinsic parameters with respect to the world reference. Then, the LIDAR–camera extrinsic calibration is formulated as a nonlinear least squares problem in terms of the constraints introduced above. It is further solved by the Levenberg-Marquardt (LM) method [11] after properly estimating the initializations. An overview of the entire calibration procedure is presented in Algorithm 1. The details are subsequently introduced below.

Algorithm 1: 3D LIDAR–camera extrinsic calibration procedure.

Input: N LIDAR–camera observations of a trihedron

Manually selected regions of the trihedron’s planes in the observations.

Output: \mathbf{R}_{LC} and \mathbf{T}_{LC}

for $i = 1 \rightarrow N$ **do**

for $j = 1 \rightarrow 3$ **do**

 | Estimate the j^{th} plane $\{\mathbf{N}_{L_i}^j, d_{L_i}^j\}$ w.r.t the i^{th} LIDAR frame;

end

 Estimate the transformation $\{\mathbf{R}_{WL_i}, \mathbf{T}_{WL_i}\}$;

end

Refine $\{\mathbf{N}_{L_i}^j, d_{L_i}^j, \mathbf{R}_{WL_i}, \mathbf{T}_{WL_i}\}$ based on all observations;

Detect features on the first image;

for $i = 2 \rightarrow N$ **do**

 | Detect and match features on the i^{th} image;

 | Estimate the transformation $\{\mathbf{R}_{C_1C_i}, \mathbf{T}_{C_1C_i}\}$;

 | Estimate $\{\mathbf{N}_{C_1}^j, d_{C_1}^j, \mathbf{N}_{C_i}^j, d_{C_i}^j\}$;

 | Estimate the transformation $\{\mathbf{R}_{WC_1}, \mathbf{T}_{WC_1}, \mathbf{R}_{WC_i}, \mathbf{T}_{WC_i}\}$;

end

Initialize \mathbf{R}_{LC} and \mathbf{T}_{LC} ;

Refine the estimates of \mathbf{R}_{LC} and \mathbf{T}_{LC} .

3.1. 3D LIDAR Extrinsic Calibration

Given the i^{th} LIDAR observation, this step is to estimate the transformation, \mathbf{R}_{WL_i} and \mathbf{T}_{WL_i} , from the world to the i^{th} LIDAR frame. To this end, we first estimate the trihedron’s planes according to the LIDAR observation. When a user specifies a set of 3D points that mostly lie on the trihedron’s j^{th} plane, the plane’s parameters $\{\mathbf{N}_{L_i}^j, d_{L_i}^j\}$ are estimated by minimizing the following linear least squares problem:

$$\arg \min_{\mathbf{N}_{L_i}^j, d_{L_i}^j} \sum_{k=1}^{M(i,j)} \|\mathbf{N}_{L_i}^j{}^T \mathbf{P}_{L_i}^{j,k} - d_{L_i}^j\|^2 \quad (6)$$

where $M(i, j)$ is the number of points on the plane. Once the three planes are determined with respect to the i^{th} LIDAR frame, \mathbf{R}_{WL_i} and \mathbf{T}_{WL_i} are computed according to the trihedral constraint given in Equations (1) and (2).

When more than one observation is available, we can further refine the results by using the planarity constraints established between each pair of the LIDAR frames. Thus, we get

$$\begin{aligned} \arg \min_{\mathbf{N}_{L_1}^j, d_{L_1}^j, \mathbf{N}_{L_i}^j, d_{L_i}^j} & \sum_{i=2}^N \sum_{k=1}^{M(i,j)} \|\mathbf{N}_{L_1}^j{}^T (\mathbf{R}_{L_1L_i} \mathbf{P}_{L_i}^{j,k} + \mathbf{T}_{L_1L_i}) - d_{L_1}^j\|^2 \\ & + \sum_{i=2}^N \sum_{k=1}^{M(1,j)} \|\mathbf{N}_{L_i}^j{}^T (\mathbf{R}_{L_1L_i} \mathbf{P}_{L_1}^{j,k} + \mathbf{T}_{L_1L_i}) - d_{L_i}^j\|^2 \end{aligned} \quad (7)$$

It is obtained by forming the first LIDAR frame with each of the remaining frames as pairs. Since $\mathbf{R}_{L_1L_i}$, $\mathbf{T}_{L_1L_i}$, $\mathbf{R}_{L_iL_1}$, and $\mathbf{T}_{L_iL_1}$ are the functions of the planes’ parameters, Equation (7) is a nonlinear

optimization problem with respect to the planes' parameters. This problem takes the previously estimated results as initializations and is solved by LM.

Let $\theta = \{\mathbf{N}_{L_1}^j, d_{L_1}^j, \mathbf{N}_{L_i}^j, d_{L_i}^j\}$ be the parameters estimated in Equation (7), and f be the function that is optimized. Then, the Levenberg–Marquardt method starts from a given initial guess θ_0 and iteratively updates the parameters via

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (8)$$

where $\Delta\theta_t$ is obtained by solving the following equation

$$(\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})) \Delta\theta_t = \mathbf{J}^T (-f(\theta_t)) \quad (9)$$

Here λ is a damping parameter determined adaptively and \mathbf{J} is the Jacobian matrix which is obtained conveniently by symbolic computation in MATLAB.

3.2. Camera Extrinsic Calibration

This step is to determine the transformations, $\{\mathbf{R}_{WC_i}, \mathbf{T}_{WC_i}\}$, from the world to the camera frames, together with the planes $\{\mathbf{N}_{C_i}^j, d_{C_i}^j\}$. In contrast to the LIDAR sensor, it is incapable of recovering all parameters from one image since no metric information is available. Hence, two LIDAR-camera observations are needed.

Given two LIDAR-camera observations, we first estimate $\mathbf{R}_{C_1C_2}$ and $\mathbf{T}_{C_1C_2}$ between the two camera frames. Once a user delimits the regions of the planes on two images, a set of point features are detected by SIFT [29] within the regions in the first image and then matched to the correspondences in the second one. The two sets of features are represented by $\{\mathbf{p}_{C_1}^k\}$ and $\{\mathbf{p}_{C_2}^k\}$, which satisfy the epipolar constraint [28]

$$\mathbf{F}^{-1}(\mathbf{p}_{C_1}^k)^T \mathbf{E} \mathbf{F}^{-1}(\mathbf{p}_{C_2}^k) = 0 \quad (10)$$

Here, \mathbf{E} is the essential matrix. The estimation of \mathbf{E} and the recovery of $\mathbf{R}_{C_1C_2}$ and $\mathbf{T}_{C_1C_2}$ from \mathbf{E} are the fundamental problems in computer vision, which are solved by the well-known eight-point algorithm [28]. However, the recovered $\mathbf{T}_{C_1C_2}$ is of unit norm. We hence use the motion constraint defined in Equation (5) to get its scale.

Once the relative motion between two views is determined, we are able to determine the planes by taking advantage of the planarity constraint established between two images, as defined in Equation (4). Hence, in the presence of noise, we estimate the pose of a plane by

$$\arg \min_{\substack{\mathbf{N}_{C_1}, d_{C_1} \\ \mathbf{N}_{C_2}, d_{C_2} \\ \mathbf{R}_{C_1C_2}, \mathbf{T}_{C_1C_2}}} \sum_{k=1}^M \left\| \mathbf{p}_{C_2}^k - \mathbf{F} \left(\mathbf{R}_{C_1C_2} \frac{d_{C_1} \mathbf{F}^{-1}(\mathbf{p}_{C_1}^k)}{\mathbf{N}_{C_1}^T \mathbf{F}^{-1}(\mathbf{p}_{C_1}^k)} + \mathbf{T}_{C_1C_2} \right) \right\|^2 + \left\| \mathbf{p}_{C_1}^k - \mathbf{F} \left(\mathbf{R}_{C_2C_1} \frac{d_{C_2} \mathbf{F}^{-1}(\mathbf{p}_{C_2}^k)}{\mathbf{N}_{C_2}^T \mathbf{F}^{-1}(\mathbf{p}_{C_2}^k)} + \mathbf{T}_{C_2C_1} \right) \right\|^2 \quad (11)$$

in which $\mathbf{R}_{C_1C_2}, \mathbf{T}_{C_1C_2}$ are also refined. It is a nonlinear least squares problem solved by LM. The estimates of \mathbf{N}_{C_i} and d_{C_i} are simply initialized with the corresponding parameters in the LIDAR frames for the simplicity. It is reasonable considering that the relative transformation between the two sensors is small when compared with those to the trihedron.

3.3. 3D LIDAR–Camera Extrinsic Calibration

With the above-estimated parameters, we now formulate the LIDAR-camera extrinsic calibration task as a nonlinear least squares problem. In terms of the planarity constraints established between the LIDAR and the camera frames, we get the form

$$\arg \min_{\mathbf{R}_{LC}, \mathbf{T}_{LC}} \sum_{i=1}^N \sum_{j=1}^3 \sum_{k=1}^{M(i,j)} \|\mathbf{N}_{c_i}^{j,T} (\mathbf{R}_{LC} \mathbf{P}_{L_i}^{j,k} + \mathbf{T}_{LC}) - d_{c_i}^j\|^2 \quad (12)$$

It is solved by LM with the initializations obtained from

$$\begin{cases} \mathbf{R}_{LC} = \mathbf{R}_{WC_i} \mathbf{R}_{WL_i}^{-1} \\ \mathbf{T}_{LC} = \mathbf{T}_{WC_i} - \mathbf{R}_{WC_i} \mathbf{R}_{WL_i}^{-1} \mathbf{T}_{WL_i} \end{cases} \quad (13)$$

with any $i = 1 \cdots N$.

4. Experiments

We implement the proposed method in MATLAB. The running time of our algorithm is coarsely measured on a laptop with an Intel Core2Duo 2.26 GHz processor and 3 GB memory. Except for the manual input procedure, it takes about 20 seconds in average to perform the entire calibration when 9 LIDAR-camera observations are considered. Each contains 5,000 LIDAR points and 100 registered image points. In order to evaluate the proposed method, a series of experiments have been carried out. The algorithm is first tested on simulated data to validate its correctness and explore its sensitivity with respect to noise. Then, it is used to calibrate a real system composed of a 3D LIDAR and a camera. The calibration results are subsequently used for 3D reconstruction.

4.1. Simulations

The first experiment validates the correctness and numerical stability of our algorithm. We hereby generate sets of data to simulate multiple observations of a trihedron obtained by a 3D LIDAR-camera system. The system is of the following properties. The rotation and translation from the LIDAR to the camera are set, respectively, as $E(\mathbf{R}_{LC}) = [11.46^\circ, 5.73^\circ, 85.94^\circ]^T$ and $\mathbf{T}_{LC} = [0.4, -0.08, 0.2]^T$ m, where $E(\mathbf{R}_{LC})$ is the Euler angle of \mathbf{R}_{LC} . An unorthogonal trihedron is synthesized, whose planes are defined by $\mathbf{N}_{C_1}^1 = [-0.342, 0.937, 0.067]^T$, $d_{C_1}^1 = -3.837$ m, $\mathbf{N}_{C_1}^2 = [-0.325, -0.930, 0.171]^T$, $d_{C_1}^2 = -7.710$ m, and $\mathbf{N}_{C_1}^3 = [0.181, 0.028, 0.983]^T$, $d_{C_1}^3 = -2.466$ m, with respect to the first camera frame. Each plane contains 5,000 LIDAR points and 100 registered image points. The simulated camera uses the following Mercator projection model:

$$\begin{cases} u = (180 - \text{atan2}(Y, X))M/360 \\ v = \text{acos}(Z/\sqrt{X^2 + Y^2 + Z^2})N/180 \end{cases} \quad (14)$$

where (u, v) denotes a pixel, and $M \times N = 1024 \times 1024$ is the resolution of an image. This projection model is one of the models that the Ladybug3 [25] camera has.

4.1.1. Performance w.r.t. the Number of Observations

The extrinsic calibration can be conducted with two or more LIDAR-camera observations. In this experiment, we investigate the impact of the observation number on the calibration performance. Nine LIDAR-camera observation pairs are generated, in which Gaussian noise with zero mean and σ standard deviation is added. We randomly select σ from the range of $[0, 0.2]$ m for LIDAR points and from the range of $[0, 1]$ pixels for image features. We vary the number of observations from 2 to 9. For each number, 200 independent trials are carried out. The estimated parameters \mathbf{R}_{LC} and \mathbf{T}_{LC} in each trial are compared with the ground truth and measured, respectively, by the displaced Euler angle of the rotation and the absolute error of the translation. Figure 2 plots the mean and standard deviation of the errors.

Figure 2. Errors vs. the number of observations. (a) presents the translation's absolute errors in X , Y , and Z directions. (b) shows the displaced Euler angle $[\alpha, \beta, \gamma]^T$.

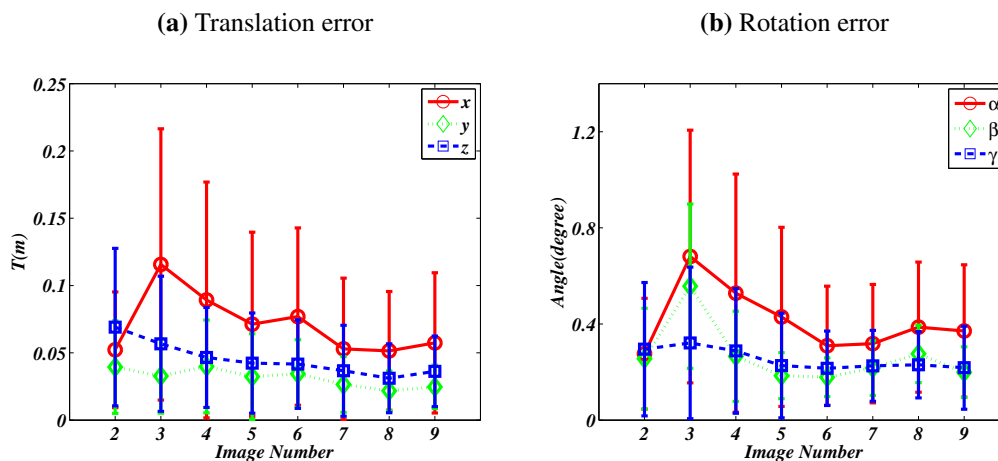


Figure 2 shows no obvious benefits achieved when the number of observations increases. The reason is that, even when we use two observations, in total there are already six planes taken into consideration. In our simulations, there is even a peak on the error corresponding to 3 observations, partly because the impact of noise is larger than the benefit achieved from the increase of observations. Hence, on the leverage of complexity and performance, throughout all the following experiments, we continue using two LIDAR-camera observations.

4.1.2. Performance w.r.t. the Noise on LIDAR Points

Real ranging sensors produce noisy measurements. Hence, this experiment explores the sensitivity with respect to noise on LIDAR points. We conduct the experiment on the first two simulated observations. Zero mean Gaussian noise is added to points of the LIDAR observations, with σ varying from 0.02 to 0.2 m. Analogous to the previous case, we conduct 200 independent trials for each noise level. The errors on the translation and rotation are evaluated and plotted in Figure 3.

Figure 3. Errors vs. the noise level on LIDAR points. (a) presents the translation's absolute errors in X , Y , and Z directions. (b) shows the displaced Euler angle $[\alpha, \beta, \gamma]^T$.

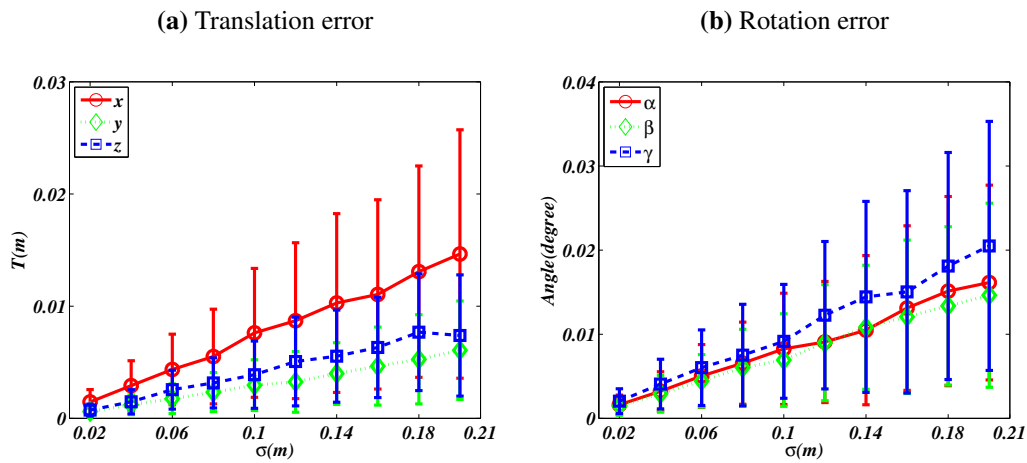


Figure 3 shows that the errors increase linearly with the noise level. When $\sigma = 0.1$ m, which is a noise level of a practical LIDAR, the translation errors are around 0.005 m in Y and Z directions, and 0.01 m in X direction. The rotation errors are about 0.01° . In our simulated configurations, X represents the direction of the optical axis, along which depth information degenerates so that larger errors are resulted in [30].

4.1.3. Performance w.r.t. the Noise on Image Points

The feature detection and matching algorithm we use in this work is SIFT [29], which is of sub-pixel accuracy. In this experiment, we investigate the sensitivity with respect to the noise on matched image features. Zero mean Gaussian noise with $\sigma \in [0.1, 1]$ pixels is added to each feature point on the first two simulated image data. Analogously to above cases, 200 trials are conducted for each noise level. The performance is evaluated and plotted in Figure 4.

Figure 4. Errors vs. the noise level on image points. (a) presents the translation's absolute errors in X , Y , and Z directions. (b) is the displaced Euler angle $[\alpha, \beta, \gamma]^T$.

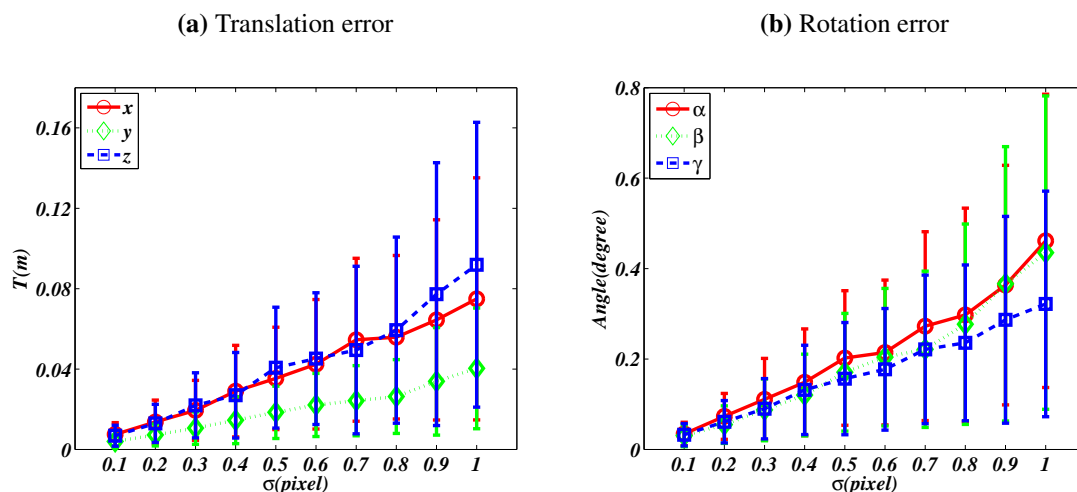
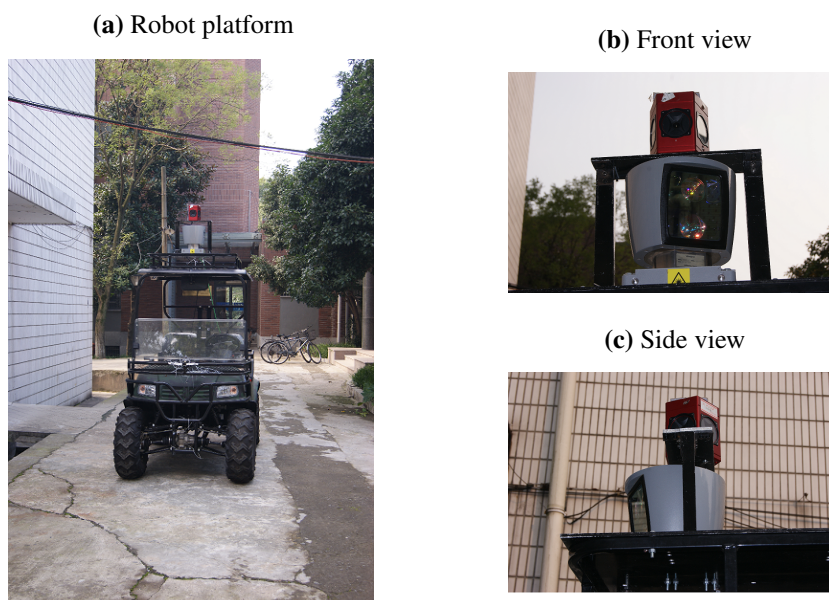


Figure 4 also presents a linear relationship between the errors and the noise level. When $\sigma = 0.5$ pixels, which is a noise level higher than the normal noise, the translation errors are smaller than 0.04 m and the rotation errors are around 0.2° .

4.2. Real Data

To further evaluate the proposed algorithm, we employ it to calibrate a real system and use the calibration results to reconstruct 3D scenes. The system is composed of a 3D Velodyne HDL-64E LIDAR [26] and a Ladybug3 spherical vision system [25], which are rigidly mounted on the roof of a vehicle, as shown in Figure 5. Both sensors produce omnidirectional measurements.

Figure 5. The robotic platform and the sensors in our experiment. (a) is the robotic platform, which is equipped with a Ladybug3 camera and a 3D Velodyne HDL-64E LIDAR, (b) shows the front view of the two sensors, and (c) is the side view.



In the experiments, we collect two LIDAR-camera measurements of a scenario containing a trihedral object. The trihedron consists of two adjacent walls of a building, together with the ground plane, as shown in Figures 6 and 7. Due to imperfect construction techniques and noise, the planes of the trihedron are not strictly orthogonal to each other. During the calibration procedure, regions of the planes are manually marked out on both LIDAR and image data, and features on the imaged trihedrons are detected and matched by SIFT [29]. A portion of the matched feature points are shown in Figure 8. Table 1 lists the calibration results of our method. For the purpose of comparison, we also include the results obtained by the checkerboard pattern-based method [12] using six observations. From the results, it is difficult to determine which one is more accurate, since no ground truth is available. Our proposed method, however, is more convenient, as it is easier to collect calibration data and requires less manual input.

In order to validate the calibration results, the determined extrinsic parameters are further used for 3D reconstruction. With the calibrated \mathbf{R}_{LC} and \mathbf{T}_{LC} , 3D LIDAR points in a view are first transformed into

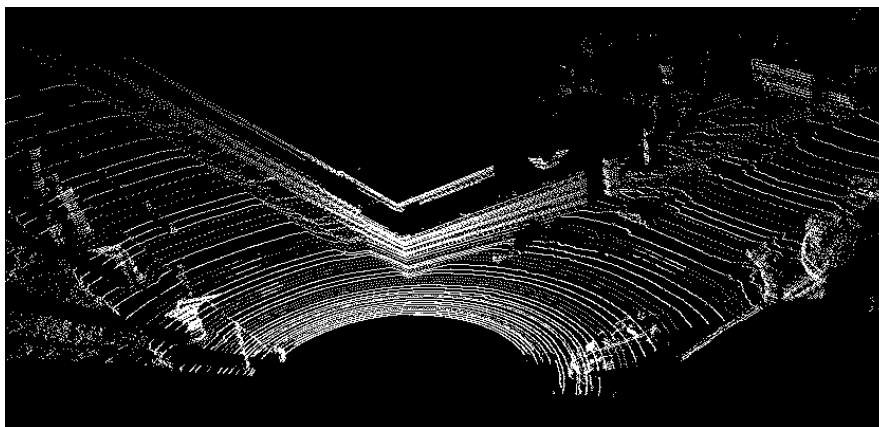
the camera frame and then registered to the image. The colors of registered image pixels are taken to render the corresponding upsampled LIDAR points. Figure 6(c) and Figure 7(c) demonstrate the colored 3D scenes of the two calibration views (only the data within a 180° field of view are shown for a better visibility), from which we see that the walls and the bushes are reconstructed well.

Figure 6. The first LIDAR-camera view used for calibration. (a) is the panoramic image captured by a Ladybug3 camera, and (b) is the 3D point cloud collected by a Velodyne HDL-64E LIDAR. (c) shows the reconstructed 3D scene.

(a) Panoramic image



(b) 3D point cloud



(c) Colored 3D point cloud

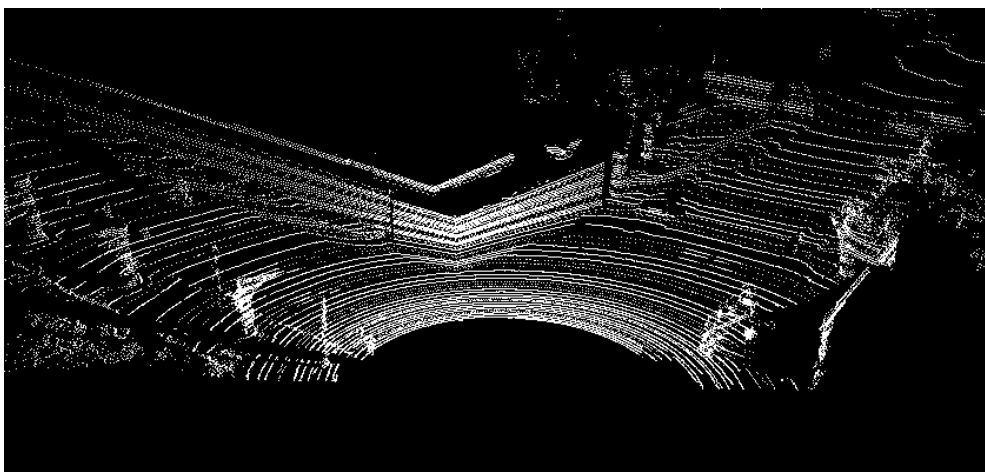


Figure 7. The second LIDAR-camera view used for calibration. (a) is the panoramic image captured by a Ladybug3 camera, and (b) is the 3D point cloud collected by a Velodyne HDL-64E LIDAR. (c) shows the reconstructed 3D scene.

(a) Panoramic image



(b) 3D point cloud



(c) Colored 3D point cloud



Figure 8. A portion of matched features on the trihedron. The matched feature pairs on the three planes are marked with lines in different styles.

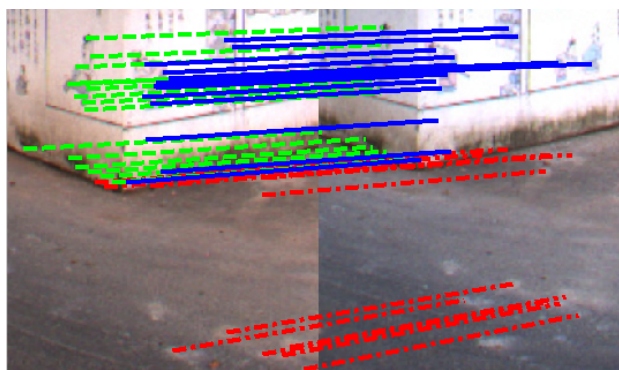


Table 1. Calibration results of a real 3D LIDAR-camera system.

	T_X (m)	T_Y (m)	T_Z (m)	α (deg)	β (deg)	γ (deg)
The proposed method	0.257	0.007	-0.323	-1.788	1.446	-88.542
The method in [12]	0.203	0.036	-0.285	-1.358	1.799	-88.996

5. Conclusions

In this paper, we have presented a new method of conducting the extrinsic calibration for a 3D LIDAR-camera system. Specifically, instead of using planar checkerboard patterns, we take advantage of arbitrary trihedral objects, which might be either orthogonal or not, for calibration. This kind of configuration is ubiquitous in structured environments, so that it is very convenient for a mobile robot to collect data. We have validated the algorithm on both simulated and real scenarios. Although the experimental results are presented from 3D LIDAR and omnidirectional camera systems, the algorithm is applicable to systems composed of any kind of 3D LIDARs and cameras. Our method is interesting for both indoor or outdoor mobile robots equipped with such sensors. The calibration results can be further used for data fusion applications.

Acknowledgements

This research work is partially supported by the National Natural Science Foundation of China via grants 61001171, 60534070, 90820306, and Chinese Universities Scientific Fund.

References

1. Ess, A.; Leibe, B.; Schindler, K.; Van, G.L. Moving Obstacle Detection in Highly Dynamic Scenes. In *Proceeding of IEEE International Conference Robotics and Automation*, Kobe, Japan, 12–17 May 2009.

2. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceeding of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012.
3. Wulf, O.; Arras, K.; Christensen, H.; Wagner, B. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proceeding of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18–22 April 2004; Volume 4, pp. 4204–4209.
4. Pandey, G.; McBride, J.R.; Eustice, R. Ford campus vision and lidar data set. *Int. J. Robot. Res.* **2011**, *30*, 1543–1552.
5. Zhao, G.; Xiao, X.; Yuan, J. Fusion of Velodyne and Camera Data for Scene Parsing. In *Proceeding of 15th International Conference on Information Fusion (FUSION)*, Seattle, WA, USA, 6–9 July 2009.
6. Wasielewski, S.; Strauss, O. Calibration of a Multi-Sensor System Laser Rangefinder/Camera. In *Proceedings of the Intelligent Vehicles Symposium*, Detroit, MI, USA, 25–26 September 1995; pp. 472–477.
7. Naroditsky, O.; Iv, E.P.; Daniilidis, K. Automatic Alignment of a Camera with a Line Scan Lidar System. In *Proceedings of IEEE International Conference Robotics and Automation*, Shanghai, China, 9–13 May 2011.
8. Zhang, Q.; Pless, R. Extrinsic Calibration of a Camera and Laser Range Finder (Improve Camera Calibration). In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Sendai, Japan, 28–30 September 2004; pp. 2301–2306.
9. Mei, C.; Rives, P. Calibration between a Central Catadioptric Camera and a Laser Range Finder for Robotic Applications. In *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, 15–19 May 2006; pp. 532–537.
10. Zhang, Z. A flexible new technique for camera calibration. *IEEE Tran. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334.
11. Madsen, K.; Nielsen, H.; Tingleff, O. Methods for Non-Linear Least Squares Problems. In *Informatics and Mathematical Modelling*; Technical University of Denmark: Lyngby, Denmark, 2004.
12. Unnikrishnan, R.; Hebert, M. *Fast Extrinsic Calibration of a Laser Rangefinder to a Camera*; Carnegie Mellon University, Robotics Institute: Pittsburgh, PA, USA, 2005.
13. Pandey, G.; McBride, J.; Savarese, S.; Eustice, R. Extrinsic Calibration of a 3D Laser Scanner and an Omnidirectional Camera. In *Proceedings of 7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy, 6–8 September 2010.
14. Rodriguez, F.S.; Fremont, V.; Bonnifait, P. Extrinsic Calibration between a Multi-Layer Lidar and a Camera. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea, 20–22 August 2008; pp. 214–219.
15. Klimentjew, D.; Hendrich, N.; Zhang, J. Mult Sensor Fusion of Camera and 3D Laser Range Finder for Object Recognition. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Salt Lake City, UT, USA, 5–7 September 2010; pp. 236–241.

16. Aliakbarpour, H.; Nuez, P.; Prado, J.; Khoshhal, K.; Dias, J. An Efficient Algorithm for Extrinsic Calibration between a 3D Laser Range Finder and a Stereo Camera for Surveillance. In *Proceedings of International Conference on Advanced Robotics*, Munich, Germany, 22–26 June 2009; pp. 1–6.
17. Pandey, G.; McBride, J.R.; Savarese, S.; Eustice, R.M. Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Toronto, Canada, 22–26 July 2012.
18. Stamos, I.; Liu, L.; Chen, C.; Wolberg, G.; Yu, G.; Zokai, S. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *Int. J. Comput. Vis.* **2008**, *78*, 237–260.
19. Mirzaei, F.M.; Kottas, D.G.; Roumeliotis, S.I. 3D lidar-camera intrinsic and extrinsic calibration: Observability analysis and analytical least squares-based initialization. *Int. J. Robot. Res.* **2012**, *in press*.
20. Geiger, A.; Moosmann, F.; Car, O.; Schuster, B. A Toolbox for Automatic Calibration of Range and Camera Sensors Using a Single Shot. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, 14–18 May 2012.
21. Scaramuzza, D.; Harati, A.; Siegwart, R. Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 29 October–2 November 2007; pp. 4164–4169.
22. Li, G.; Liu, Y.; Dong, L.; Cai, X.; Zhou, D. An Algorithm for Extrinsic Parameters Calibration of a Camera and a Laser Range Finder Using Line Features. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 29 October–2 November 2007; pp. 3854–3859.
23. Mirisola, L.G.B.; Lobo, J.; Dias, J. 3D Map Registration using Vision/Laser and Inertial Sensing. In *Proceedings of European Conference on Mobile Robots (EMCR)*, Freiburg, Germany, 19–21 September 2007.
24. Nuez, P.; Jr, P.D.; Rocha, R.; Dias, J. Data Fusion Calibration for a 3D Laser Range Finder and a Camera Using Inertial Data. In *Proceedings of European Conference on Mobile Robots (ECMR)*, Dubrovnik, Croatia, 23–25 September 2009; pp. 31–36.
25. Grey, P. Ladybug3 Getting Started Manual, 2012. Available online: <http://www.ptgrey.com/> (accessed on 25 January 2013).
26. Velodyne. Velodyne HDL-64E, 2012. Available online: <http://velodynelidar.com/lidar> (accessed on 25 January 2013).
27. Ly, S.; Démonceaux, C.; Vasseur, P. Translation Estimation for Single Viewpoint Cameras Using Lines. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 3–8 May 2010; pp. 1928–1933.
28. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
29. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.

30. Kinoshita, K. 3D Accuracy Improvement from an Image—Evaluation and Viewpoint Dependency. In *Proceedings of Australia-Japan Advanced Workshop on Computer Vision*, Adelaide, Australia, 9–11 September 2003.

© 2013 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 license (<http://creativecommons.org/licenses/by/3.0/>).