

Article

# Simultaneous Calibration of Odometry and Head-Eye Parameters for Mobile Robots with a Pan-Tilt Camera

Nachaya Chindakham <sup>1</sup>, Young-Yong Kim <sup>2</sup> and Alongkorn Pirayawaraporn <sup>1</sup>  
and Mun-Ho Jeong <sup>1,\*</sup> 

<sup>1</sup> Department of Control and Instrumentation Engineering; Kwangwoon University, Kwangwoon-ro 1-gil 60, Nowon-gu, Seoul 01890, Korea

<sup>2</sup> Research and Development Department; Thinkware Visual Technology, 240 Pangyoyeok-ro, Bundang-gu, Gyeonggi-do, Seongnam-si 463-400, Korea

\* Correspondence: mhjeong@kw.ac.kr; Tel.: +82-010-2940-2353

Received: 13 June 2019; Accepted: 10 August 2019; Published: 20 August 2019



**Abstract:** In the field of robot navigation, the odometric parameters, such as wheel radii and wheelbase length, and the relative pose of the optical sensing camera with respect to the robot are very important criteria for accurate operation. Hence, these parameters are necessary to be estimated for more precise operation. However, the odometric and head-eye parameters are typically estimated separately, which is an inconvenience and requires longer calibration time. Even though several researchers have proposed simultaneous calibration methods that obtain both odometric and head-eye parameters simultaneously to reduce the calibration time, they are only applicable to a mobile robot with a fixed camera mounted, not for mobile robots equipped with a pan-tilt motorized camera systems, which is a very common configuration and widely used for wide view. Previous approaches could not provide the z-axis translation parameter between head-eye coordinate systems on mobile robots equipped with a pan-tilt camera. In this paper, we present a full simultaneous mobile robot calibration of head-eye and odometric parameters, which is appropriate for a mobile robot equipped with a camera mounted on the pan-tilt motorized device. After a set of visual features obtained from a chessboard or natural scene and the odometry measurements are synchronized and received, both odometric and head-eye parameters are iteratively adjusted until convergence prior to using a nonlinear optimization method for more accuracy.

**Keywords:** mobile robot kinematics; odometry calibration; head-eye calibration; simultaneous mobile robot calibration

## 1. Introduction

Robot navigation is one of the key challenges facing the field of mobile robotics. This is because mobile robots are required to drive themselves through a given environment using the information gathered from their sensors. These sensors include proprioceptive sensors such as motor-speed sensors, wheel-load sensors, joint-angle sensors, battery-voltage sensors and the inertial measurement unit (IMU), which produces a type of data called odometric data. Additionally, these robots are fitted with exteroceptive sensors such as image-feature sensors, distance sensors, light-intensity sensors, sound-amplitude sensors, and global positioning system (GPS) sensors. However, robot localization is typically inaccurate due to the uncertainty associated with measurement errors during robot configuration. Although the robot configuration data, such as wheel radii and wheelbase length, can be obtained simply from the robot specifications or by manual measurement, the actual parameters can be dissimilar in practice. This is due to systematic errors such as manufacturing errors, assembly errors, tire pressure variations, and load variations that reduce the precision associated with the

movement of the mobile robot. It is therefore necessary to estimate these odometric parameters to improve the robot's operational precision.

Antonelli's calibration method [1] uses the least-squares technique to observe a linear mapping between the unknowns and the measurements. It aims to identify a 4-parameter model, while the modified version [2] estimates the physical odometric parameters and yields a 3-parameter model without any requirement for a predefined path. Some researchers, such as [3,4], have reduced the cumulative error in odometry by considering the coupled effect of errors in diameter, wheelbase errors, and scaling errors using a popular odometry calibration method for wheeled mobile robots, developed at and named the University of Michigan Benchmark test (UMBmark) [5]. Some trials required a static wireless sensor network and GPS devices to be equipped on mobile robots to correct for the odometry errors [6,7]. However, this approach is disadvantageous in terms of cost-effectiveness.

Nowadays, mobile robots equipped with cameras which provide single color images, stereo images, or depth images are widely used in the fields of robot navigation, reconstruction, and mapping. Using this information, the robot can perform more precise and varied tasks. However, the relationship between the camera and robot in terms of 3D position and orientation is also important for accurate operation. Shiu and Ahmad [8] introduced a solution for rigid transformation between the sensor and the robot, calculated in the form  $AX = XB$ . Hand-eye calibration, as proposed by Tsai and Lenz [9], is a similar, albeit more efficient solution which does not depend on the number of images. In mobile robots, the 3D position and orientation of the camera relative to the robot's base is considered instead. Kim et al. [10] found that the head-eye transformation between the robot's coordinate system and the camera's coordinate system can be estimated simply and accurately by using the minimum variance technique, which is resistant to noisy environments.

In practical applications, the odometric parameters, such as wheel radii and wheelbase length, and the head-eye parameters are typically estimated separately, which requires a longer calibration time and increased inconvenience due to redundancy in these methods. To avoid the disadvantages, Antonelli et al. [11] have proposed a simultaneous calibration method that performs both odometry and the head-eye calibrations simultaneously. Since the method required only the synchronized measurement of odometric data and visual features, it was successful in terms of reducing the calibration time and improving efficiency in the mobile robot calibration. However, their approach is only applicable to the mobile robot on which a fixed camera is mounted, while recent mobile robots are equipped with pan-tilt motorized camera systems for wide view. This caused incomplete estimation of the head-eye parameters, that is, the method could not provide the z-axis translation parameter. Shusheng Bi et al. [12] presented an improved version of [11] in terms of accuracy, but still did not overcome the problem. Hengbo Tang et al. [13] solved the problem by taking the advantage of the planar constraints of the landmarks. Despite their accurate estimation of the head-eye parameters and the odometric parameters, there is a clear limitation in the sense that a very constrained environment is needed and several recognizable landmarks must be premeasured and fixed.

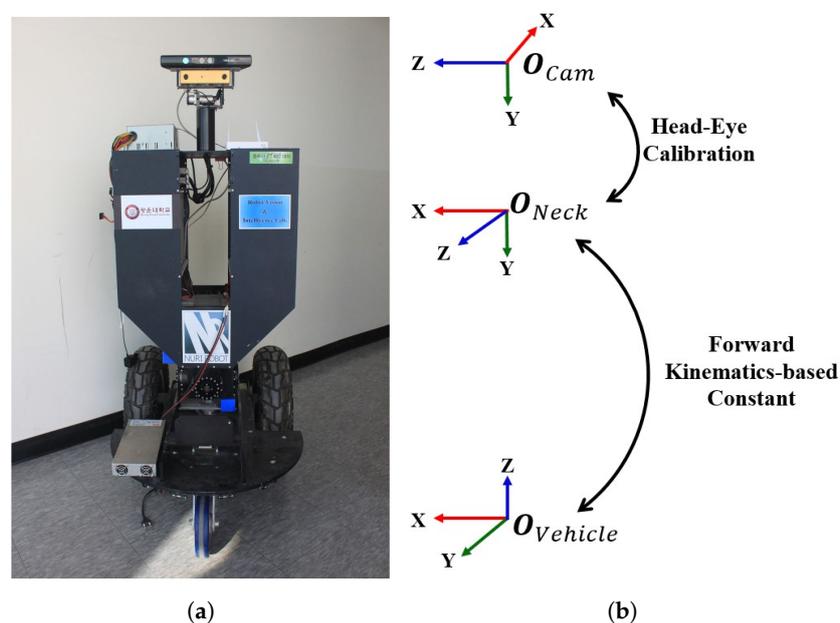
In this paper, we present a full mobile robot calibration of head-eye and odometric parameters, building on [14]. The full six parameters (rotation and translation) in the head-eye calibration and the three parameters (wheel radii and wheelbase length) in the odometry calibration are simultaneously estimated. The mobile robot equipped with a mono or stereo camera moves while the camera mounted on the pan-tilt motorized device is capturing chessboards or natural scenes. After simply planned robot movements, the full mobile robot calibration algorithm is performed using both odometry measurements and visual features such as chessboard's corners or natural feature points from a stereo camera, which are obtained by Speeded-Up Robust Feature (SURF) [15]. The head-eye and odometric parameters are iteratively adjusted to obtain the values, which are searched as a good starting point close to the ground truth please confirm intended meaning is retained. and then finally fine-tuned with the direct search-based optimization as Powell's method [16]. The remainder of this paper is organized as follows: Section 2 describes our mobile robot configuration and the relationship between each joint. Section 3 proposes an iterative-based calibration method for mobile robot having a camera

mounted on a motorized neck. Section 4 presents our experimental results, and the conclusion is summarized in Section 5.

## 2. Mobile Robot Configuration

### 2.1. Robot Coordinate System

The mobile robot configuration in Figure 1a can be understood in terms of an overview of the coordinate system of a mobile robot with a pan-tilt camera, as depicted in Figure 1b. It consists of the vehicle coordinate (robot's base),  $O_{Vehicle}$ ; the neck coordinate includes a pan-tilt joint, ( $O_{Neck}$ ), with a camera mounted on the top as camera coordinate,  $O_{Cam}$ . The relation between the vehicle system  $O_{Vehicle}$  and  $O_{Neck}$  is estimated using Denavit–Hartenberg parameters (DH parameters) depending on the mobile robot configuration. The rotation from the neck to camera and the rotations between robot's bases are estimated in Section 3.2. The remain parameters are calculated in Section 3.4.



**Figure 1.** Mobile robot configuration. (a) Robot having a pan-tilt neck equipped with a camera (front-view); (b) coordinate system of mobile robot configuration (side-view).

### 2.2. Robot Wheel Parameters

In the field of robot navigation, one of the important parameters for mobile robot calibration is the wheel parameters, which are the radii of the left and right robot wheels and the baseline (the axle length between left and right wheel), as shown in Figure 2. The robot kinematics can be expressed as

$$\begin{aligned} \dot{x} &= v \cos(\theta), \\ \dot{y} &= v \sin(\theta), \\ \dot{\theta} &= \omega, \end{aligned} \quad (1)$$

where  $v$ ,  $\omega$ , and  $\theta$  are the velocity, angular velocity, and the orientation of the mobile robot, respectively, as depicted in Figure 2. These parameters can be obtained by using the following equation:

$$\begin{aligned} v &= \frac{r_R}{2} \omega_R + \frac{r_L}{2} \omega_L, \\ \omega &= \alpha_R \omega_R + \alpha_L \omega_L, \end{aligned} \quad (2)$$

where  $\alpha_R = \frac{r_R}{b}$  and  $\alpha_L = -\frac{r_L}{b}$ . The wheel parameters as  $r_R$ ,  $r_L$ , and  $b$  are right and left wheel radii and the length of the baseline, respectively.  $\omega_R$  and  $\omega_L$  are angular velocities, which are calculated using the encoder on the right and left wheels. The ratio between the radii of wheels and the length of wheelbase is represented in terms of intermediate parameters,  $(\alpha_R, \alpha_L)$ , which are used in the calibration process instead of the real wheel parameters.

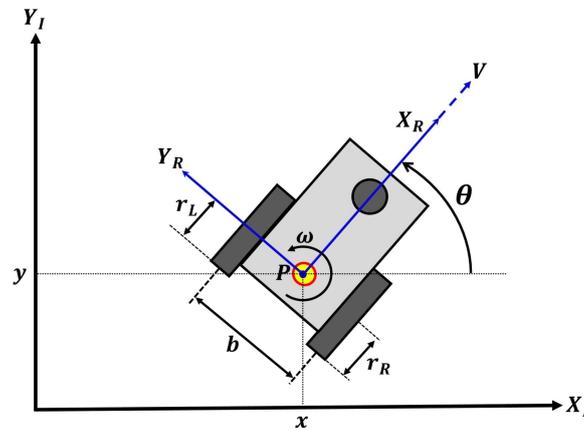


Figure 2. Mobile robot odometry and its relevant variables.

The rotational angle of the mobile robot from frame  $i$  to frame  $j$ , by  $t_i = 0$  and  $t_j = t$ , can be obtained through the integral of Equation (2) with respect to time as follows:

$$\begin{aligned}\theta(t) &= \alpha_R \int_0^t \omega_R(\tau) d\tau + \alpha_L \int_0^t \omega_L(\tau) d\tau, \\ &= \alpha_R \phi_R(t) + \alpha_L \phi_L(t),\end{aligned}\quad (3)$$

where  $\phi_R(t)$ ,  $\phi_L(t)$  are the encoder positions of the right and left wheels, respectively.

### 3. Simultaneous Calibration for Mobile Robot with Pan-Tilt Camera

In this section, we describe the proposed calibration method separately in six parts. The closed-loop transformation of the camera and the robot base to the robot neck between any frames  $i$  and  $j$ , where  $i = 1, \dots, N - 1$  and  $j = i + 1$  are concisely depicted in Section 3.1. Since a set of captured images,  $I_{n=1}^{n=N}$ , and the calibration data set (such as rotating angles of wheels ( $\phi_{R_i}, \phi_{L_i}$ ), and transformation from robot base coordinate to robot's neck coordinate,  $T_{N_i}^{V_i}$ ) are obtained once as the calibration input data. Sections 3.2–3.4 explain how to use these data to obtain the head-eye rotation, intermediate wheel parameters, and head-eye translation, including the actual wheel parameters, respectively. These processes are estimated iteratively until the value of all parameters converge, as described in Section 3.5. Finally, Section 3.6 describes the non-linear optimization method that increases the accuracy of the calibration results.

#### 3.1. Closed-Loop Transformations

Let us now consider the homogeneous transformation of different vehicle poses from frame  $i$  to  $j$ , as shown in Figure 3. The abbreviations  $O_C, O_N, O_V$  are the camera, neck, and vehicle coordinate systems, respectively. The closed-loop diagram can be represented by Equation (4).

$$T_{N_i}^{V_i} T_{C_i}^{N_i} T_{C_j}^{C_i} = T_{V_j}^{V_i} T_{N_j}^{V_j} T_{C_j}^{N_j}, \quad (4)$$

where  $T_N^V$  is the physical relationship between  $O_{Neck}$  and  $O_{Vehicle}$ . The head-eye homogeneous transformation is  $T_C^N$ . When the camera is directed at the same target or feature, camera motion

is represented as  $T_{C_j}^{C_i}$ . The robot's motion between frame  $i$  and  $j$  is presented to  $T_{V_j}^{V_i}$ . The homogeneous transformation of Equation (4) can be decomposed into rotational and translational terms as follows:

$$R_{N_i}^{V_i} R_{C_i}^{N_i} R_{C_j}^{C_i} = R_{V_j}^{V_i} R_{N_j}^{V_j} R_{C_j}^{N_j} \tag{5}$$

$$R_{N_i}^{V_i} R_{C_i}^{N_i} t_{C_j}^{C_i} + R_{N_i}^{V_i} t_{C_i}^{N_i} + t_{N_i}^{V_i} = R_{V_j}^{V_i} R_{N_j}^{V_j} t_{C_j}^{N_j} + R_{V_j}^{V_i} t_{N_j}^{V_j} + t_{V_j}^{V_i} \tag{6}$$

where  $R$  is the  $3 \times 3$  rotation matrix and  $t$  is  $3 \times 1$  translation vector. Equation (5) is used to obtain the head-eye rotation parameters,  $R_C^N$ , which consist of 3 degrees of freedom (DOF),  $(\gamma_x, \gamma_y, \gamma_z)$ . Equation (6) refers to the translation of the system, which is used to estimate the head-eye translation  $(t_{C,x}^N, t_{C,y}^N, t_{C,z}^N)$  and the actual size of the wheel parameters  $(r_R, r_L, b)$ . These parameters are calculated in Section 3.4.

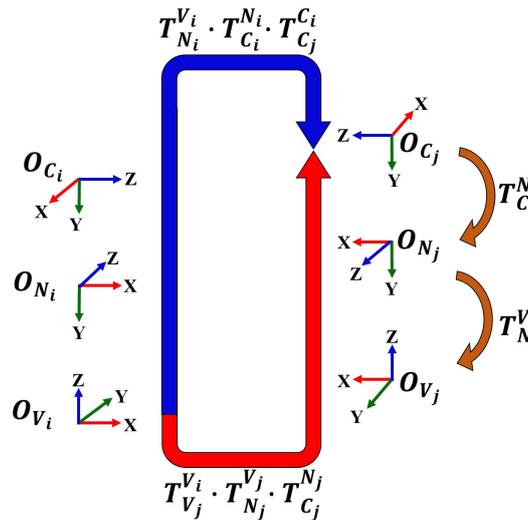


Figure 3. Closed-loop transformation between any frame  $i$  and  $j$ .

### 3.2. Head-Eye Rotation Estimation

The six parameters in head-eye calibration consisting of three for rotation and another three for the translation, which are necessary and required to be obtained before an operation. In this section, three parameters of the rotation between the robot's neck and the camera are calculated precisely. A set of visual measurements and robot movement data were obtained by moving the robot and capturing images synchronously and continuously. The rotation between the camera and the robot's base was estimated accurately, as by Antonelli et al. [11]. They obtained the rotation parameters between the robot's base and camera using Equivalent angle-axis representation. However, their approach achieved because their mobile robot had a camera equipped on a fixed neck that did not change the relation between the robot's base and camera. If the relation between the base and the camera was changed during the calibration data collection, the Equivalent angle-axis method could not be used to solve the problem. During collecting the input calibration data of our mobile robot, both the robot and its neck move that means the rotation between the base and the camera are also changed.

In fact, whenever the robot's neck moves around the pan-tilt axis, the coordinate of the camera mounted on that neck is also moved significantly. Therefore, the relationship between the camera and the robot's neck is static, which means the subscript  $i$  of  $R_{N_i}^{C_i}$  can be omitted as  $R_N^C = R_{N_1}^{C_1} = R_{N_2}^{C_2} = \dots = R_{N_N}^{C_N}$ , where  $N$  is the total number of input images. Moreover, the mobile robot rotation is performed only on a planar. In other words, the mobile robot rotates around z-axis only [11]. Hence,  $R_{V_j}^{V_i}$ , in Equation (5) can be replaced with  $R_z(\theta)$ . Assuming the rotation matrices  $R_{C_j}^{C_i}$ ,  $R_{V_i}^{N_i}$ , and  $R_{V_j}^{V_i}$  are

known, which are thoroughly described in Section 4, an Equation (5) can be represented in form of  $AX = XB$ , and the rotation  $R_N^C$  can be obtained likewise [8,9] as follows:

$$R_{C_j}^{C_i} R_N^C = R_N^C R_{V_i}^{N_i} R_z(\theta) R_{N_j}^{V_j}, \tag{7}$$

where the matrix  $X$  is the estimated head-eye rotation,  $R_N^C$ . The camera rotations,  $R_{C_j}^{C_i}$ , are represented to matrix  $A$ . The remaining variables of the right side  $R_{V_i}^{N_i} R_z(\theta) R_{N_j}^{V_j}$  are demonstrated to matrix  $B$ , which  $i = 1, \dots, N - 1$  and  $j = 2, \dots, N$ .

### 3.3. Intermediate Wheel Parameters Estimation

In this section, the linear relationships of intermediate wheel parameters,  $(\alpha_R, \alpha_L)$ , and rotational angle of the robot movement from the previous to current positions prior to capture any image  $i, \theta_j$ , which is obtained with Equation (3) using the period time of the robot movement between the previous and current positions, which is  $\theta(1) = 0$  as no loss of generality. The change in the rotational angle of the robot’s base from frames  $i$  to  $j$ , on a planar  $\theta(t_j)$  is also re-estimated. According to the robot’s movement on a plane, the change in rotational angle about the  $z$ -axis of the robot’s base coordinates, which is assumed to be perpendicular to the floor, between a pair of consecutive frames is  $R_z(\theta)$ . In practical applications, the  $z$ -axis at the robot’s base coordinates from one consecutive frame to another, ( $V_i$  and  $V_j$ ), may not be parallel because of an error in the estimated  $R_C^N$  and the floor plane of any pair of positions are not parallel. Therefore, the rotational angle about the  $z$ -axis at the robot’s base coordinates between frames  $i$  and  $j$  can be calculated using the Euler angle (ZYX) as follows:

$$\theta_j = \theta(t_j) = \text{Atan2}(r_{21}, r_{11}), \quad R_{V_j}^{V_i} = R_{N_i}^{V_i} R_C^N R_{C_j}^{C_i} R_N^C R_{N_j}^{V_j}, \tag{8}$$

where  $r_{21}$  and  $r_{11}$  are generic elements of  $R_{V_j}^{V_i}$ . If we consider for  $N$  images, the representation of Equation (3) can be used to obtain the parameters,  $(\alpha_R, \alpha_L)$ , similar to [11], as follows:

$$\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_N \end{bmatrix} = \begin{bmatrix} \phi_{\theta_1} \\ \vdots \\ \phi_{\theta_N} \end{bmatrix} \begin{bmatrix} \alpha_R \\ \alpha_L \end{bmatrix} = \bar{\phi}_\theta \begin{bmatrix} \alpha_R \\ \alpha_L \end{bmatrix}, \tag{9}$$

where  $\phi_{\theta_j} = [\phi_R(t_j) \quad \phi_L(t_j)]$ , are obtained from the rotational angles of both wheels from position  $i$  to  $j$ , ( $i = 1, \dots, N - 1$  and  $j = 2, \dots, N$ ).  $\bar{\phi}_\theta$  is a matrix with  $N \times 2$  dimensions. The intermediate wheel parameters,  $(\alpha_R, \alpha_L)$ , can be calculated using the linear least squares method as follows

$$\begin{bmatrix} \alpha_R \\ \alpha_L \end{bmatrix} = (\bar{\phi}_\theta^T \bar{\phi}_\theta)^{-1} \bar{\phi}_\theta^T \cdot \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_N \end{bmatrix} \tag{10}$$

### 3.4. Head-Eye Translation and Wheel Parameters Estimation

The head-eye rotation and intermediate parameters have already been obtained in Sections 3.2 and 3.3. The remaining parameters that are estimated in this section are  $t_C^N$ , the head-eye translation vector, and the actual wheel parameters  $r_R, r_L, b$ . The translational and the rotational components of the robot’s base coordinates can be described using the mobile robot kinematic equations as

$$\begin{aligned} x_j &= x_i + \tau v_i \cos(\theta_i + \tau \omega_i / 2), \\ y_j &= y_i + \tau v_i \sin(\theta_i + \tau \omega_i / 2), \\ \theta_j &= \theta_i + \tau \omega_i, \end{aligned} \tag{11}$$

where  $i$  and  $j$  denote the previous and current frames, respectively, and  $\tau$  is the period of time between frames. Substituting the intermediate wheel parameters and Equation (2) into Equation (11) yields

$$\begin{aligned} x_j &= x_i + \tau \left( -\frac{\alpha_R}{2\alpha_L} r_L \omega_{R,i} + \frac{r_L}{2} \omega_{L,i} \right) \cos(\theta_i + \tau\omega_i/2), \\ y_j &= y_i + \tau \left( -\frac{\alpha_R}{2\alpha_L} r_L \omega_{R,i} + \frac{r_L}{2} \omega_{L,i} \right) \sin(\theta_i + \tau\omega_i/2). \end{aligned} \quad (12)$$

In fact,  $t_{V_j}^i$  is the relative translation in  $x$  and  $y$  directions, which can be rewritten as

$$\begin{aligned} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} &= \frac{\tau}{2} \begin{bmatrix} (-\frac{\alpha_R}{\alpha_L} \omega_{R,i} + \omega_{L,i}) \cos(\theta_i + \tau\omega_i/2) \\ (-\frac{\alpha_R}{\alpha_L} \omega_{R,i} + \omega_{L,i}) \sin(\theta_i + \tau\omega_i/2) \end{bmatrix} r_L, \\ t_{V_j}^i &= \begin{bmatrix} \lambda_1 & \lambda_2 & 0 \end{bmatrix}^T \cdot r_L, \end{aligned} \quad (13)$$

From Equation (13), the translation between two robot base positions can be substituted into Equation (6) representing the relationship between the coordinates as follows:

$$(R_{N_i}^{V_i} - R_z(\theta) R_{N_j}^{V_j}) t_C^N - \begin{bmatrix} \lambda_1 & \lambda_2 & 0 \end{bmatrix}^T \cdot r_L = -R_{N_i}^{V_i} R_C^N t_{C_j}^i - t_{N_i}^{V_i} + R_z(\theta) t_{N_j}^{V_j}, \quad (14)$$

where  $R_{N_i}^{V_i} R_C^N$  and  $(R_{N_i}^{V_i} - R_z(\theta) R_{N_j}^{V_j})$  are referred to the matrices  $A$  and  $B$ , respectively. Thus, Equation (14) can be simplified as

$$B t_C^N - \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 0 \end{bmatrix} r_L = -A t_{C_j}^i - t_{N_i}^{V_i} + R_z(\theta) t_{N_j}^{V_j}, \quad (15)$$

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} t_{C,x}^N \\ t_{C,y}^N \\ t_{C,z}^N \end{bmatrix} - \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 0 \end{bmatrix} r_L = - \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} t_{C_j,x}^i \\ t_{C_j,y}^i \\ t_{C_j,z}^i \end{bmatrix} - \begin{bmatrix} t_{N,x}^V \\ t_{N,y}^V \\ t_{N,z}^V \end{bmatrix} + \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{N,x}^V \\ t_{N,y}^V \\ t_{N,z}^V \end{bmatrix} \quad (16)$$

from Equation (16), the third component of the vector  $[\lambda_1 \ \lambda_2 \ 0]^T$  is zero, which also makes the third row zero. It can be derived as follows:

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} & -\lambda_1 \\ B_{21} & B_{22} & B_{23} & -\lambda_2 \\ B_{31} & B_{32} & B_{33} & 0 \end{bmatrix} \begin{bmatrix} t_{C,x}^N \\ t_{C,y}^N \\ t_{C,z}^N \\ r_L \end{bmatrix} = \begin{bmatrix} A_{11} t_{C_j,x}^i + A_{12} t_{C_j,y}^i + A_{13} t_{C_j,z}^i + t_{N,x}^V - C\theta t_{N,x}^V + S\theta t_{N,y}^V \\ A_{21} t_{C_j,x}^i + A_{22} t_{C_j,y}^i + A_{23} t_{C_j,z}^i + t_{N,x}^V - S\theta t_{N,x}^V - C\theta t_{N,y}^V \\ r_L \end{bmatrix}. \quad (17)$$

Consider Equation (17) over all frames. The final equation can be expressed as

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} & -\lambda_1 \\ B_{21} & B_{22} & B_{23} & -\lambda_2 \\ \vdots & \vdots & \vdots & \vdots \\ B_{2N1} & B_{2N2} & B_{2N3} & -\lambda_{2N} \end{bmatrix} \begin{bmatrix} t_{C,x}^N \\ t_{C,y}^N \\ t_{C,z}^N \\ r_L \end{bmatrix} = \begin{bmatrix} A_{11} t_{C_j,x}^i + A_{12} t_{C_j,y}^i + A_{13} t_{C_j,z}^i + t_{N,x}^V - C\theta t_{N,x}^V + S\theta t_{N,y}^V \\ A_{21} t_{C_j,x}^i + A_{22} t_{C_j,y}^i + A_{23} t_{C_j,z}^i + t_{N,x}^V - S\theta t_{N,x}^V - C\theta t_{N,y}^V \\ \vdots \\ A_{2N1} t_{C_j,x}^i + A_{2N2} t_{C_j,y}^i + A_{2N3} t_{C_j,z}^i + t_{N,x}^V - S\theta t_{N,x}^V - C\theta t_{N,y}^V \end{bmatrix}, \quad (18)$$

where the left matrix of the left term has  $2N \times 4$  dimensions; and the right term is a vector with  $2N$  dimensions.  $N$  is a total number of frames,  $N = 1, \dots, 2N$ . From Equation (18), the head-eye translation and the actual radii of the left wheel,  $(t_{C,x}^N, t_{C,y}^N, t_{C,z}^N, r_L)$ , are estimated using a linear least-squares method. The remaining parameters  $r_R$  and  $b$  can be obtained with Equation (2). The wheel parameters,  $(r_L, r_R, b)$ , which are obtained in this section, will be used to re-estimate the rotation between the robot's head and neck, as previously described.

### 3.5. Linearly Iterative Estimation

Even though the previous approach [11] could estimate the odometric and head-eye parameters simultaneously, their method could not provide the completed six parameters. The translation of z-axis between head-eye coordinates,  $t_{C,z}^N$  was not obtained by their method. The proposed method presents a fully mobile robot calibration of odometric and head-eye parameters, building on [14]. The rotation and translation in head-eye calibration including  $t_{C,z}^N$  and the odometric parameters are simultaneously estimated precisely. Supposing that the values of all parameters are not obtained correctly before optimization, it leads all parameters to convergence with incorrect values or divergence. Therefore, this section explains our contribution that we apply iteration-based estimation to initially guess the values of all parameters correctly that leads all parameters to convergence with the correct values rapidly. The processes of Sections 3.2–3.4 are used to compute repeatedly until all parameters are converged. The head-eye parameters,  $R_N^C$ , are estimated from Section 3.2 prior to being used to calculate the intermediate wheel parameters,  $(\alpha_R, \alpha_L)$ , of both wheels in Section 3.3. After that, they are used to compute the remaining parameters, as described in Section 3.4. These results are also used again to calculate the head-eye parameters following Section 3.2, as shown in Algorithm 1, which shows steps 3 to 5 compute repeatedly until convergence.

---

**Algorithm 1** Full algorithm simultaneous calibration for mobile robot with pan-tilt camera.

---

**Input:**  $\{T_{N_i}^{V_i}\}_{i=1}^{i=N}$ ,  $\{\phi_{R_i}, \phi_{L_i}\}_{i=1}^{i=N}$ ,  $I_{i=1}^{i=N}$

**Output:**  $R_N^C$ ,  $t_C^N$ ,  $r_L$ ,  $r_R$ ,  $b$

**for**  $i \in 1, \dots, N - 1$  **do**

**Step 1:** Obtain  $T_{C_j}^{C_i}$  between each frame using chessboard's corners or natural features

**end for**

**Step 2:** Initial  $r_L, r_R, b$  with manual measurements and obtain  $\theta$  with Equation (3)

**while** Convergence **do**

**Step 3:** Compute  $R_N^C$  with Equation (7)

**Step 4:** Compute  $\alpha_R, \alpha_L$  with Equations (8)–(10)

**Step 5:** Compute  $t_C^N, r_L, r_R, b$  with Equation (18)

**end while**

**Step 6:** Refine  $R_N^C, t_C^N, r_L, r_R, b$  with Equations (19)

---

### 3.6. Non-Linear Optimization

Even though all parameters that are estimated in the previous section can lead to a good initial estimation, they are probably not the correct and accurate values. Therefore, a method of minimizing a function as Powell's method [16] is applied to fine-tune all parameters as closely as the ground truth. The variables consisting of the Euler angles of the head-eye rotation,  $R_N^C$ ,  $(\gamma_x, \gamma_y, \gamma_z)$ , head-eye translation,  $(t_{C,x}^N, t_{C,y}^N, t_{C,z}^N)$ , and wheel parameters,  $(r_L, r_R, b)$  are refined using the following equation:

$$\Theta^* = \arg \min_{R_N^C, t_C^N, r_L, r_R, b} C(\Theta),$$

$$C(\Theta) = \sum_{i=1}^{N-1} \sum_{k=1}^K [Q_{i,k} - Q'_{i,k}]^2, \quad (19)$$

$$Q'_{i,k} = T_N^C T_{V_i}^{N_i} T_{V_j}^{V_i} T_{N_j}^{V_j} T_C^N [X_j \ Y_j \ Z_j \ 1]^T, \quad j = i + 1,$$

where  $Q'_{i,k}$  is the predicted 3D features, which are used to transform any point  $k$  from frame  $j$ ,  $j = i + 1$ , to frame  $i$ . The 3D feature at frame  $i$ , represented with  $Q_{i,k}$  and  $T_C^N$ , is constructed using head-eye rotational and translational parameters,  $(\gamma_x, \gamma_y, \gamma_z, t_{C,x}^N, t_{C,y}^N, t_{C,z}^N)$ .  $T_N^V$  is calculated using DH parameters and pan-tilt data from the encoders; while  $T_{V_j}^V$  is referred to  $R_z(\theta)$ , which is calculated using wheel encoder data and the estimated wheel parameters,  $r_L, r_R, b$ .  $N$  and  $K$  are the total number of images and features, respectively.

#### 4. Experimental and Results

In our experiments, we used a mobile robot, which had two RGB cameras mounted on the pan-tilt motorized device, as shown in Figure 1. A chessboard and natural scenes were used as the calibration target for single camera and a pair of cameras (a stereo camera), respectively. The differential-drive mobile robot moved to any specified position and both the robot's neck and pan-tilt axis also moved before capturing an image sequentially. The mobile robot moved and captured repeatedly to obtain a set of images,  $I_{n=1}^{n=N}$ . The data of the robot movements such as the rotational angles of both wheels ( $\phi_{R_n}, \phi_{L_n}$ ) and of any image  $n$  ( $n = 1, \dots, N$ ) were obtained by their angular velocity ( $\omega_{R_n}, \omega_{L_n}$ ) and the movement period time,  $\tau_i$ , as shown in Equation (3). The angles of wheels at the starting position,  $n = 1$ , were determined with  $\phi_{R_1} = \phi_{L_1} = 0$ . The transformation, including rotation and translation between robot's base and robot's neck coordinate systems,  $T_{N=1, \dots, N}^{V_{n=1, \dots, N}}$ , were calculated with the rotational angles of the pan-tilt axis at the robot's neck and Denavit–Hartenberg parameters (DH parameters).

In the case that used a chessboard as the calibration target, we captured a set of images using single camera with a resolution of  $320 \times 240$  pixels. The chessboard contained  $10 \times 7$  black and white square grids (56 corner points), and the size of any grid was  $5.4 \times 5.4$  cm. We extracted the feature points of the chessboard's corners manually using [17]. Since all feature points of the chessboard's corners were obtained, the transformation  $T_{C_i}^W$  between the camera at position  $i$  and the chessboard, which was determined to be the world coordinate, can be estimated with a plane-based transformation estimation [18]. Therefore, the transformation between any pair of camera positions  $i$  and  $j$ ,  $T_{C_j}^{C_i}$ , was simply estimated with  $T_{C_j}^{C_i} = T_W^{C_i} T_{C_j}^W$ .

In the case of natural scenes, the natural features were observed from the real environment based on rectified images. The corresponding feature points of the stereo images were estimated with SURF [19]. The transformations between any pair of camera positions,  $T_{C_j}^{C_i}$ , were obtained by a closed-form solution of the least-squares problem of absolute orientation using orthonormal matrices [20]. The result of stereo matching is shown in Figure 4.

The transformations ( $T_C^N$  and  $T_C^V$ ), odometric parameters, and 3D back-projection error results of the proposed and Antonelli's methods [11] were compared in Table 1. In the table of  $T_C^N$ , the head-eye rotation parameters ( $\gamma_x, \gamma_y, \gamma_z$ ) were obtained by the ZYX-Euler angle corresponding to the rotation matrix  $R_C^N$ . The comparison of the head-eye transformation result indicated that the rotation and translation calibrated with the proposed method showed completed parameters estimation, while Antonelli's method did not obtain the transformation between the camera and the robot neck due to the fact that their mobile robot's neck could not move.

Furthermore, we also compared the results of the transformation between the robot's base and the camera,  $T_C^V$ , by calculation of the transformation at the starting position,  $T_{C_1}^{V_1}$ , which was obtained by  $T_{N_1}^{V_1}$  and  $T_C^N$ . The transformation matrix,  $T_{C_1}^{V_1}$ , of the proposed method was similar to Antonelli's except the translation,  $t_{C,z}^V$ , that Antonelli's method could not provide due to the constraining of the origin of the vehicle reference frame on the inertial x-y plane. The error was a 3D back-projection error, which was calculated with the average of Euclidean distance from all 3D feature points between any image and the transformed 3D feature points of other images, which was shown in mm units. Figure 5 also presents the reprojection result between frames of our method.



Figure 4. Natural features matching.

Table 1. Results of the proposed method and Antonelli's method.

Coordinate	Unit	Parameter	Proposed Method	Antonelli's Method
$T_C^N$	Deg.	$\gamma_x$	−45.5715	-
		$\gamma_y$	−82.0160	-
		$\gamma_z$	45.4678	-
	mm.	$t_{C,x}^N$	31.5182	-
		$t_{C,y}^N$	−75.1629	-
		$t_{C,z}^N$	−90.0127	-
$T_C^V$	Deg.	$\gamma_x$	78.6669	−62.4582
		$\gamma_y$	−84.3044	−87.9764
		$\gamma_z$	11.0493	153.6164
	mm.	$t_{C,x}^V$	321.9080	221.4345
		$t_{C,y}^V$	−89.4305	−50.3979
		$t_{C,z}^V$	969.8691	<i>unknown</i>
wheel	mm.	$r_L$	202.4040	225.3074
		$r_R$	200.6111	227.3912
		$b$	490.4046	513.1268
Error	mm.		4.4239	7.9798

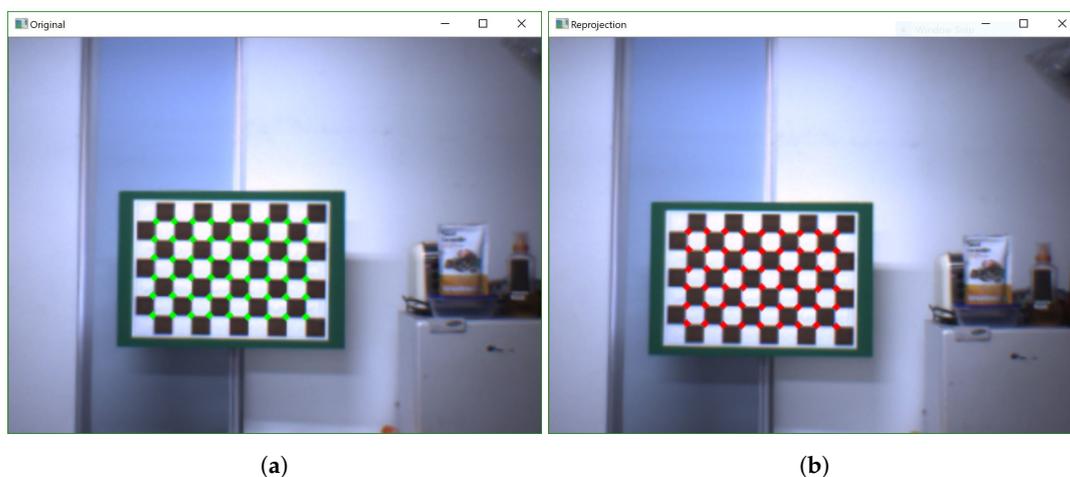
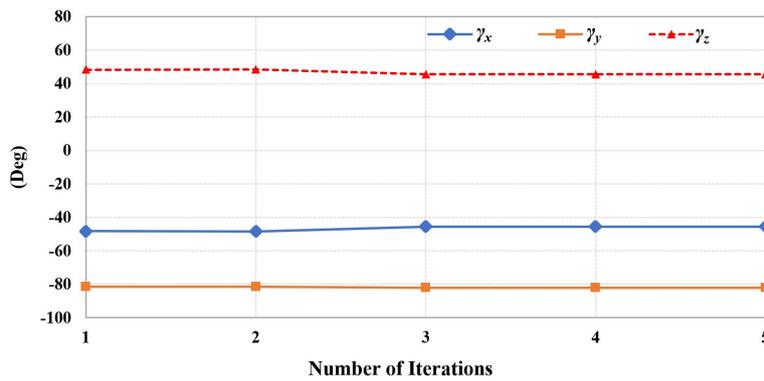


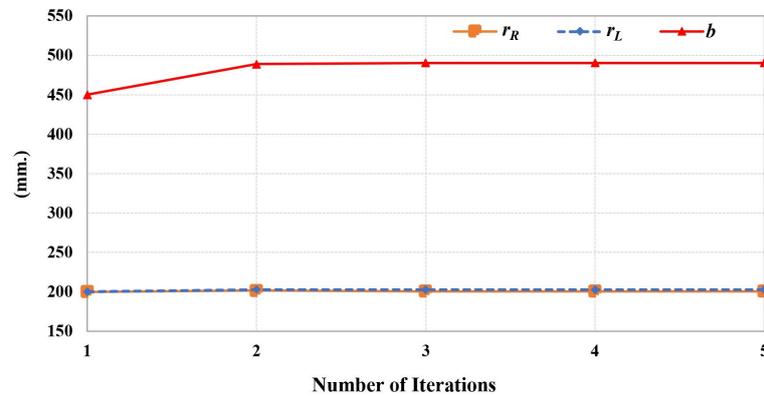
Figure 5. Reprojection results: (a) Reprojection of image  $i$ ; (b) transformed reprojection of image  $j$ .

Even though our method requires an iterative computation in Section 3.5, all parameters reach stability within just a few iterations, as shown in Figure 6. The calibration error after optimization

using both chessboard and natural scenes with respect to the number of iterations is shown in Figure 7. However, the 3D back-projection error of the calibration using natural scenes also depends on accuracy of the stereo matching process. Although the back-projection error using natural features is significantly higher than using features from a chessboard, both cases required only a few optimized iterations before the error was steady, which demonstrated the back-projection error of 4.4239 mm, as represented in Table 1. The back-projection error before optimization (iteration = 0) and after optimization of both chessboard and natural scenes related to the number of images are represented in Figure 8. It shows the number of poses that affect to the calibration accuracy. However, the required number of input images using a chessboard as the calibration target is at least 30 input images, while using the natural features requires at least 35 input images for the steady results.



(a) Head-eye parameters



(b) wheel parameters

Figure 6. Rate of change related to number of iterative estimation.

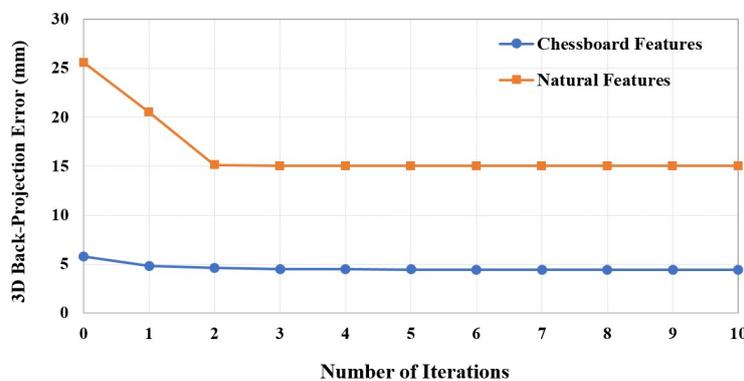


Figure 7. 3D back-projection error after optimization related to number of iterations.

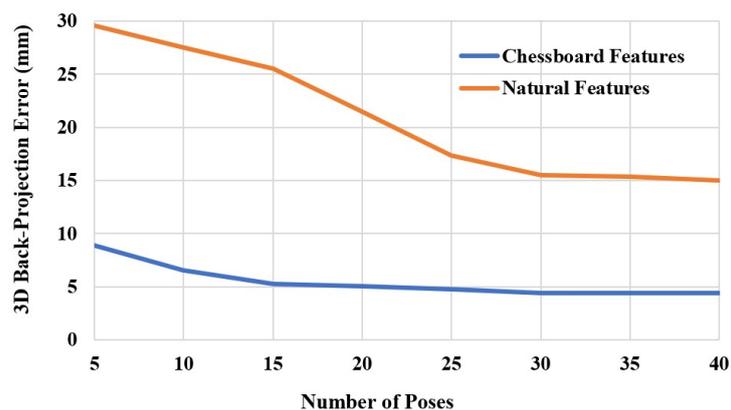


Figure 8. 3D back-projection error after optimization related to number of poses.

## 5. Conclusions

In this paper, we presented an approach for simultaneous calibration of head-eye and odometric parameters on the mobile robot equipped with a camera mounted on the motorized pan-tilt. Our proposed approach involves complete estimation of the wheel radii, wheelbase length, and the rotation and translation of the head-eye. Additionally, we obtain comprehensive results of the relative pose between the camera and the robot's base, showing that our proposed method can compute the translation in  $z$ -axis while the previous studies could not. After the data from the visual features of either chessboard's corners or natural scenes and odometry measurements were acquired, both head-eye and wheel parameters were simultaneously estimated by using iterative adjustment until all parameters converged—the experimental results showed a few iterations were necessary for the convergence. Furthermore, nonlinear optimization is used to minimize the cost function to more sufficiently and appropriate to perform on the mobile robot equipped with a pan-tilt camera precisely.

**Author Contributions:** Conceptualization, N.C., Y.-Y.K. and M.-H.J.; Methodology, N.C., Y.-Y.K. and A.P.; Software & Validation, N.C. and A.P.; Writing-Original Draft Preparation, N.C. and Y.-Y.K.; Writing-Review & Editing, M.-H.J., N.C. and A.P.; Project Administration & Funding Acquisition, M.-H.J.

**Funding:** This journal was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology(2017R1D1A1B03035700).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Antonelli, G.; Chiaverini, S.; Fusco, G. A calibration method for odometry of mobile robots based on the least-squares technique: Theory and experimental validation. *IEEE Trans. Robot.* **2005**, *21*, 994–1004. [[CrossRef](#)]
2. Antonelli, G.; Chiaverini, S. Linear estimation of the physical odometric parameters for differential-drive mobile robots. *Auton. Robot.* **2007**, *23*, 59–68. [[CrossRef](#)]
3. Lee, K.; Jung, C.; Chung, W. Accurate calibration of kinematic parameters for two wheel differential mobile robots. *J. Mech. Sci. Technol.* **2011**, *25*, 1603–1611. [[CrossRef](#)]
4. Bei, X.; Ping, X.; Gao, W. Calibration of systematic errors for wheeled mobile robots. *Int. J. Sci. Eng. Sci.* **2017**, *1*, 14–16.
5. Borenstein, J.; Feng, L. UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots. In Proceedings of the SPIE Conference on Mobile Robots, Philadelphia, PA, USA, 22–26 October 1995.

6. Shenoy, S.; Tan, J. Simultaneous Localization and Mobile Robot Navigation in a Hybrid Sensor Network. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005.
7. Song, G.; Zhuang, W.; Wei, Z.; Song, A. An Effective Algorithm for Guiding Mobile Nodes in Wireless Sensor Networks. In Proceedings of the IEEE Workshop on Signal Processing Systems, Shanghai, China, 17–19 October 2007.
8. Shiu, Y.C.; Ahmad, S. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form  $AX = XB$ . *IEEE Trans. Robot. Autom.* **1989**, *1*, 16–29. [[CrossRef](#)]
9. Tsai, R.Y.; Lenz, R.K. A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. *IEEE Trans. Robot. Autom.* **1989**, *5*, 345–358. [[CrossRef](#)]
10. Kim, S.-J.; Jeong, M.-H.; Lee, J.-J.; Lee, J.-Y.; Kim, K.-G.; You, B.-J.; Oh, S.-R. Robot head-eye calibration using the minimum variance method. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Tianjin, China, 14–18 December 2010; pp. 1446–1451.
11. Antonelli, G.; Caccavale, F.; Grossi, F.; Marino, A. Simultaneous calibration of odometry and camera for a differential drive mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010; pp. 5417–5422.
12. Bi, S.; Yang, D.; Cai, Y. Automatic Calibration of Odometry and Robot Extrinsic Parameters Using Multi-Composite-Targets for a Differential-Drive Robot with a Camera. *Sensors* **2018**, *18*, 3097. [[CrossRef](#)] [[PubMed](#)]
13. Tang, H.; Liu, Y. A Fully Automatic Calibration Algorithm for a Camera Odometry System. *IEEE Sens. J.* **2017**, *17*, 4208–4216. [[CrossRef](#)]
14. Young, Y.K.; Jeong, M.-H.; Kang, D.J. Mobile robot calibration. In Proceedings of the 13th International Conference on Control, Automation and Systems, Gwangju, Korea, 20–23 October 2013; pp. 694–697.
15. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [[CrossRef](#)]
16. Powell, M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **1964**, *7*, 155–162. [[CrossRef](#)]
17. Bouquet, J. Camera Calibration Toolbox for Matlab. From Computational Vision at CALTECH. Available online: [http://www.vision.caltech.edu/bouquetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouquetj/calib_doc/index.html) (accessed on 14 October 2015).
18. Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999.
19. Goh, K.M.; Mokji, M.M.; Abu-Bakar, S.A.R. SURF Based Image Matching from Different Angle of Viewpoints using Rectification and Simplified Orientation Correction. *World Acad. Sci. Eng. Technol.* **2012**, *6*, 852–856.
20. Horn, B.K.P.; Hilden, H.M.; Negahdaripour, S. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A* **1988**, *5*, 1127–1135. [[CrossRef](#)]



© 2019 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/>).