

Article

## Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments

I-Hsum Li <sup>1</sup>, Ming-Chang Chen <sup>2</sup>, Wei-Yen Wang <sup>3,\*</sup>, Shun-Feng Su <sup>2</sup> and To-Wen Lai <sup>3</sup>

<sup>1</sup> Department of Information Technology in Lee-Ming Institute of Technology, No.2-2, Lizhuan Rd., Taishan Dist., New Taipei 243, Taiwan; E-Mail: i-hsum@orion.ee.ntust.edu.tw

<sup>2</sup> Department of Electrical Engineering, National Taiwan University of Science and Technology, 43 Keelung Road, Section 4, Taipei 106, Taiwan;

E-Mails: maximchen.chen@gmail.com (M.-C.C.); sfsu@mail.ntust.edu.tw (S.-F.S.)

<sup>3</sup> Department of Applied Electronics Technology, National Taiwan Normal University, 160, He-ping East Rd., Section 1, Taipei 106, Taiwan; E-Mail: blanc131@gmail.com

\* Author to whom correspondence should be addressed; E-Mail: wywang@ntnu.edu.tw; Tel.: +886-2-7734-3536; Fax: +886-2-2351-5092.

Received: 10 September 2013; in revised form: 22 January 2014 / Accepted: 23 January 2014 /

Published: 27 January 2014

---

**Abstract:** A single-webcam distance measurement technique for indoor robot localization is proposed in this paper. The proposed localization technique uses webcams that are available in an existing surveillance environment. The developed image-based distance measurement system (IBDMS) and parallel lines distance measurement system (PLDMS) have two merits. Firstly, only one webcam is required for estimating the distance. Secondly, the set-up of IBDMS and PLDMS is easy, which only one known-dimension rectangle pattern is needed, *i.e.*, a ground tile. Some common and simple image processing techniques, *i.e.*, background subtraction are used to capture the robot in real time. Thus, for the purposes of indoor robot localization, the proposed method does not need to use expensive high-resolution webcams and complicated pattern recognition methods but just few simple estimating formulas. From the experimental results, the proposed robot localization method is reliable and effective in an indoor environment.

**Keywords:** indoor robot localization; image-based distance measurement system; parallel lines distance measurement system

---

## 1. Introduction

Autonomous robots have a wide range of potential applications in security guards, house cleaning and even warfare. Most of them are equipped with position measurement systems (PMSs) for the purpose of precisely locating themselves and navigating in their working fields. Three typical techniques [1] in PMSs are triangulation, scene analysis, and proximity. The triangulation technique uses the geometric properties of triangles to compute object locations. The most well-known technique is the Global Positioning System (GPS). However, GPS, as it is satellite dependent, has an inherent problem of accurately determining the locations of objects within a building [2]. A proximity location-sensing technique entails determining when an object is “near” a known location, and the object’s presence can be sensed via some limited range physical phenomenon. Some famous techniques are detecting physical contact [3,4] or monitoring wireless cellular access points [5,6]. The scene analysis location sensing technique uses features of a scene observed from a particular vantage point to draw conclusions about the location of the observer or of objects in the scene. Some well-known techniques are a radar location system [7] or a visual images location system [8].

In an indoor localization technique, the infrared light [6], ultrasonic [9], laser range finder [10,11], RFID [12], and radar [13] are the most popular wireless techniques. Diffuse infrared technology is commonly used to realize indoor locations, but the short-range signal transmission and line-of-sight requirements limit the growth. Ultrasonic localization [9] uses the time-of-flight measurement technique to provide location information. However, the use of ultrasound requires a great deal of infrastructure in order for it to be highly effective and accurate. Laser distance measurement is executed by measuring the time that it takes for a laser light to be reflected off a target and returned back to the sender. Because the laser range finder is a very accurate and quick measurement device, this device is widely used in many applications. In [10,11], Subramanian *et al.* and Barawid *et al.* proposed an autonomous vehicle guidance system based on a laser rangefinder. The laser rangefinder was used to acquire environment distance information that can be used to identify and avoid obstacles during navigation. In [14], Thrun *et al.* provided an autonomous navigation method based on a particle filter algorithm. In this study, the laser rangefinder can receive all the measurement information that it can utilize to compute the likelihood of the particles. These papers confirm that laser rangefinders are high performance and high accuracy measurement equipment. However, their high performance relies on high hardware costs. RFID-based localization uses RF tags and a reader with an antenna to locate objects, but the detection of each tag only can work over approximately 4 to 6 meter distances. To improve the low precision on location positioning, the well-known SpotON [15] technology uses an aggregation algorithm based on radio signal strength analysis for 3D-location sensing. However, a complete system is not available yet. An RF-based RADAR system [7,16,17] uses the 802.11 network adapter to measure signal strengths at multiple base stations positioned to provide overlapping coverage for locating and tracking objects inside buildings. Unfortunately, most cases to date cannot provide overall accuracy of systems as optimal as desired. In indoor localization for robots, most of these wireless techniques are used to perform scans of static obstacles around the robots, and the localization is calculated by matching those scans with a metric map of the environment [18,19], but in dynamic environments the detected static-features are often not enough for estimating a robust localization.

Li *et al.* [20] proposed a NN-based mobile phone localization technique using Bluetooth connectivity. In this large-scale network, mobile phones equipped with GPS represent beacons, and others could connect to the beacon phones with Bluetooth connectivity. By formulating the Bluetooth network as an optimization problem, a recurrent neural network is developed to distributively find the solutions in real time. However, in general, the sampling rate of Bluetooth is relatively low, and then accurately estimating a moving object in real time is not easy. In [21], a recurrent neural network was proposed to search a desirable solution for a range-free localization of WSNs under the condition that the WSNs can be formed as a class of nonlinear inequalities defined on a graph. Taking advantage of parallel computation of the NN, the proposed approach can effectively solve the WSN localization problem, although the limited transmission bandwidth might cause difficulty in the localization.

Recently, image-based techniques have been preferred over wireless techniques [4,5,9,22]; this is because they are passive sensors and are not easily disturbed by other sensors. In [1], a portable-PC capable of marker detection, image sequence matching, and location recognition was proposed for an indoor navigation task. JongBae *et al.* used the augmented reality (AR) technique to achieve an average location recognition success rate of 89%, though the extra cost must be considered in this technique. In [23], Cheoket *et al.* provided a method of localization and navigation in wide indoor areas with a wearable computer for human-beings. Though the set-up cost is lower, this method is not easy to implement and set up if users do not know the basic concept of electronic circuit analysis and design. Furthermore, an imaged-based method for distance measurement was proposed in [24–29]. According to the transform equations in those papers, the distance can be calculated from the ratio of the size between the pre-defined reference points and the measured object. In recent years, we have seen growing importance placed on research in two-camera localization systems [30,31]. From two different images, the object distances can be calculated by a triangular relationship. However, to ensure the measuring reliability, the photography angle and the distance between two cameras must be maintained at the same position. Due to the use of two cameras for the measuring device, the set-up costs of the experimental environment will be increased.

Nowadays, surveillance systems exist in most modern buildings, and cameras have been configured around these buildings. In general, one camera covers one specific area. In order to locate an autonomous patrolling robot using existing cameras in buildings, a single-camera localization technique must be developed for the patrolling robots. This study aims to develop a single-webcam distance measurement technique for indoor robot localization with the purposes of saving set-up costs and increasing the accuracy of distance measurements. In our approach, the working area setting can be as simplified as possible, because the existing webcams in the surveillance environment can be utilized without any change. For a single webcam in its working coverage area, we develop an improved image-based distance measurement system (IBDMS) and a parallel lines distance measurement system (PLDMS) to measure the location of a robot according to a known-size rectangle pattern, *i.e.*, a ground tile. This measurement system uses four points, *i.e.*, the four corners of a ground tile, to form a pair of parallel lines in the webcam image. Referring to the pair of parallel lines, we can measure the location of a robot within the visual range of a webcam. Because of the fixed monitoring area of an individual webcam, few simple image processing strategies are used to search for the robots before going through IBDMS and PLDMS. First, we use the low-pass filter and on-line background update method to reduce background noise, and adopt the image morphology to complete prospect

information and to remove the slight noise. When the mobile robot is located, IBDMS and PLDMS can obtain the real-world coordinates of a mobile robot. Finally, the localization of a mobile robot can be shown on the two-dimensional map immediately. Thus, for the purpose of indoor robot localization, the proposed method does not need to use complicate pattern recognition methods, but just few simple estimation formulas.

## 2. Photography Methods

Before locating a robot by the proposed single-webcam localization technique, the acquired images must go through photograph processing for removing noise and unnecessary information. These techniques include a gray scale, a background subtraction, a morphological image processing, and a connected components labeling technique. Next, we briefly discuss the procedures [32] of these photographic correction techniques used in this paper.

### 2.1. Camera Calibration

Distortion could happen in captured images, especially is cheap webcams are used. To attenuate distortion of the captured images and thus increase the accuracy of the robot location task, the camera calibration should be done before the localization is attempted. OpenCV has taken into account the radial and tangential factors for the image distortion problem. The radial factor can be calculated by the following equations:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2)$$

The tangential distortion can be corrected via the equations as follows:

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (3)$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \quad (4)$$

In Equations (1–4) the pixel  $(x, y)$  is the image coordinate in the input image and  $(x_{corrected}, y_{corrected})$  is the image coordinate in the corrected output image. The distortion coefficient vector can be represented as  $c_{di} = [k_1 k_2 p_1 p_2 k_3]$ . Moreover, the unit conversion can be represented as:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{before} \\ y_{before} \\ z_{before} \end{bmatrix} \quad (5)$$

where  $w$  is explained by the use of homography coordinate system (and  $w = z_{before}$ ),  $f_x$  and  $f_y$  are the camera focal length, and  $c_x$  and  $c_y$  are the optical centers expressed in pixels coordinates. After calculating the camera Equation (5) and the distortion coefficients  $c_{di}$ , the functions *initUndistortRectifyMap()* and the *remap()* can calibrate the distorted images. Figure 1a shows the images before the calibration done for three webcams, and Figure 1b shows the images after the calibration procedure. For the 1st webcam (HD Webcam C310, Logitech, Lausanne, Switzerland) in our experimental environment, the distortion coefficient vector is:

$$c_{d1} = [0.0895 \ -0.7064 \ -0.0034 \ 0.0020 \ 1.3227] \quad (6)$$

and the camera conversion matrix is:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 518.38 & 0 & 253.56 \\ 0 & 518.07 & 644.23 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{before} \\ y_{before} \\ z_{before} \end{bmatrix} \quad (7)$$

For the 2nd webcam (HD Pro Webcam C920, Logitech, Lausanne, Switzerland) in our experimental environment, the distortion coefficient vector is:

$$c_{d2} = [0.2460 \ -1.8737 \ -0.0023 \ -0.0043 \ 5.4119] \quad (8)$$

and the camera conversion matrix is:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 598.36 & 0 & 318 \\ 0 & 600.76 & 264.19 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{before} \\ y_{before} \\ z_{before} \end{bmatrix} \quad (9)$$

For the 3rd webcam (HD Webcam PC235, Ronald, Osaka, Japan) in our experimental environment, the distortion coefficient vector is:

$$c_{d3} = [0.8433 \ -12.8097 \ -0.0339 \ 0.0185 \ 111.8152] \quad (10)$$

and the camera conversion matrix is:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1272.37 & 0 & 378.46 \\ 0 & 1283.87 & 249.52 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{before} \\ y_{before} \\ z_{before} \end{bmatrix} \quad (11)$$

**Figure 1.** Calibration procedure for the webcam images. (a) Before the calibration produces for three webcams, (a.1) Logitech HD Webcam C310; (a.2) Logitech HD Pro Webcam C920; (a.3) Ronald HD Webcam PC235. (b) Images after the calibration produces for three webcams, (b.1) Logitech HD Webcam C310; (b.2) Logitech HD Pro Webcam C920; (b.3) Ronald HD Webcam PC235.



(a.1)

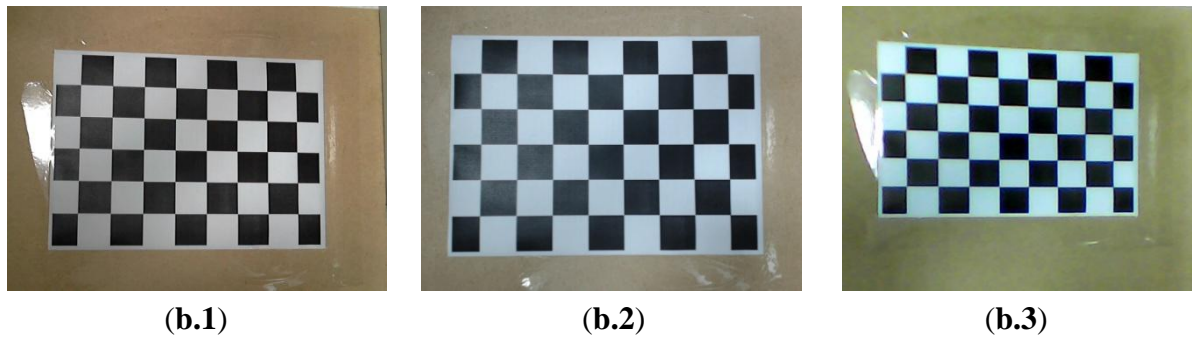


(a.2)



(a.3)

Figure 1. Cont.



## 2.2. Image Segmentation

In a grayscale image, the value of each pixel carries only intensity information. It is known as a black-and-white image, which is composed exclusively of shades of gray. Black is at the weakest intensity and white is at the strongest one. The gray scale technique can change a color image into a black-and-white image. The luminance  $f_l(x, y)$  of the  $pixel(x, y)$  is described as:

$$f_l(x, y) = 0.299R_{(x,y)} + 0.587G_{(x,y)} + 0.114B_{(x,y)} \quad (12)$$

where  $R_{(x,y)}$ ,  $G_{(x,y)}$ , and  $B_{(x,y)}$  are color values at the  $pixel(x, y)$ . Equation (12) can create binary images by a threshold value  $t$  from a grayscale image:

$$f_{binary}(x, y) = \begin{cases} 255 & \text{if } f_l(x, y) \geq t \\ 0 & \text{else} \end{cases} \quad (13)$$

Subtracting a binary acquired image  $f_{current}(x, y)$  from a binary background image  $f_{bg}(x, y)$ , we can obtain a binary foreground image as:

$$f_{fg}(x, y) = |f_{bg}(x, y) - f_{current}(x, y)| \quad (14)$$

## 2.3. Morphological Image Processing

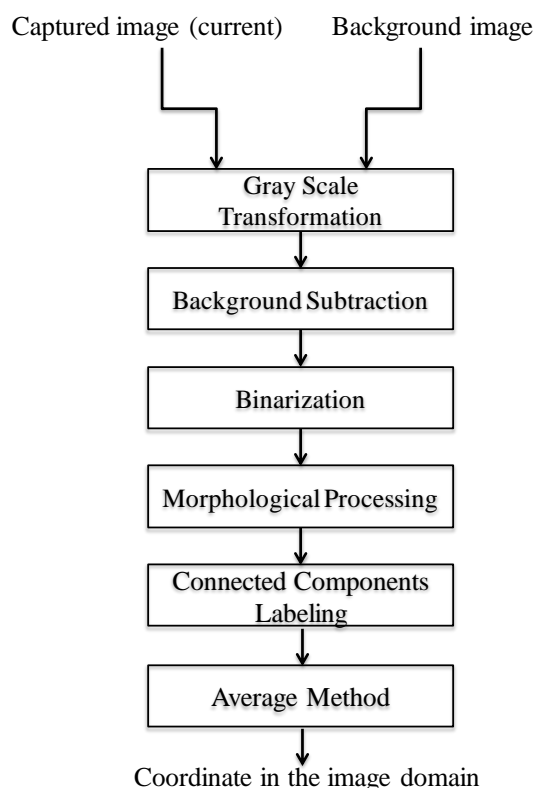
After the process of image segmentation, discontinuous edges and noise may happen in a foreground image. These will cause wrong judgments during object identification. Therefore, this paper utilizes some morphological image processing operations, such as dilation, erosion, opening and closing, in order to enable the underlying shapes to be identified and optimally reconstruct the image from their noisy precursors.

## 2.4. Connected-Components Labeling

The aim of connected-component labeling is to identify connected-components that share similar pixel intensity values, and then to connect them with each other. The connected-component labeling scans an image and groups pixels into one or more components according to pixel connectivity. Once all groups are determined, each pixel is labeled with a grey level on the basis of the component.

According to the aforementioned discussions, we can locate a robot in the captured image in an image-domain. Figure 2 shows the overall schemes of the image processing, and the experimental results are shown in Figure 3. In Figure 4,  $R_{center}$  is the center of the robot in the processed image and can be easily calculated by the simple average method. In this paper,  $R_{center}$  stands for the center-coordinate of the robot in the image-domain.

**Figure 2.** Procedures of the image processing techniques for the robot localization.



**Figure 3.** (a) Background image. (b) Captured image. (c) Gray scale of the background image. (d) Gray scale of the captured image. (e) Background subtraction. (f) Binarization processing of the foreground image. (g) Morphological processing of the foreground image. (h) Connected components labeling.

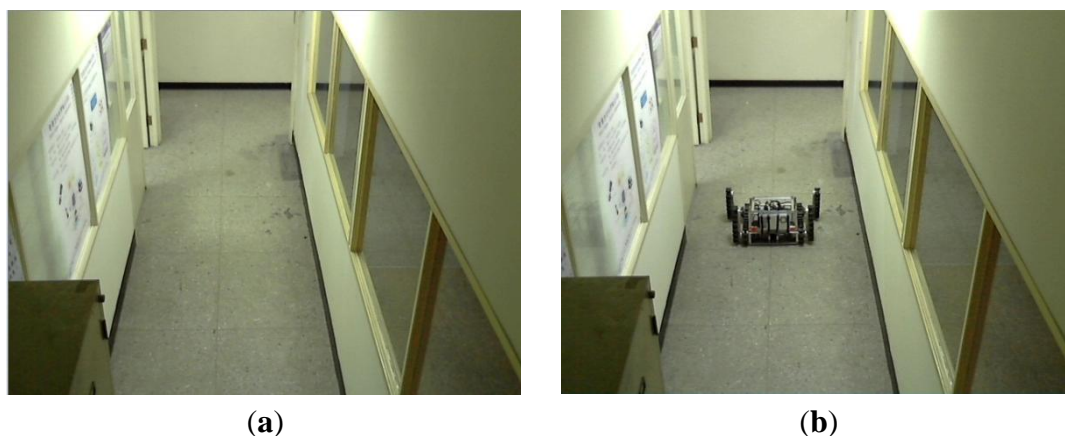


Figure 3. Cont.

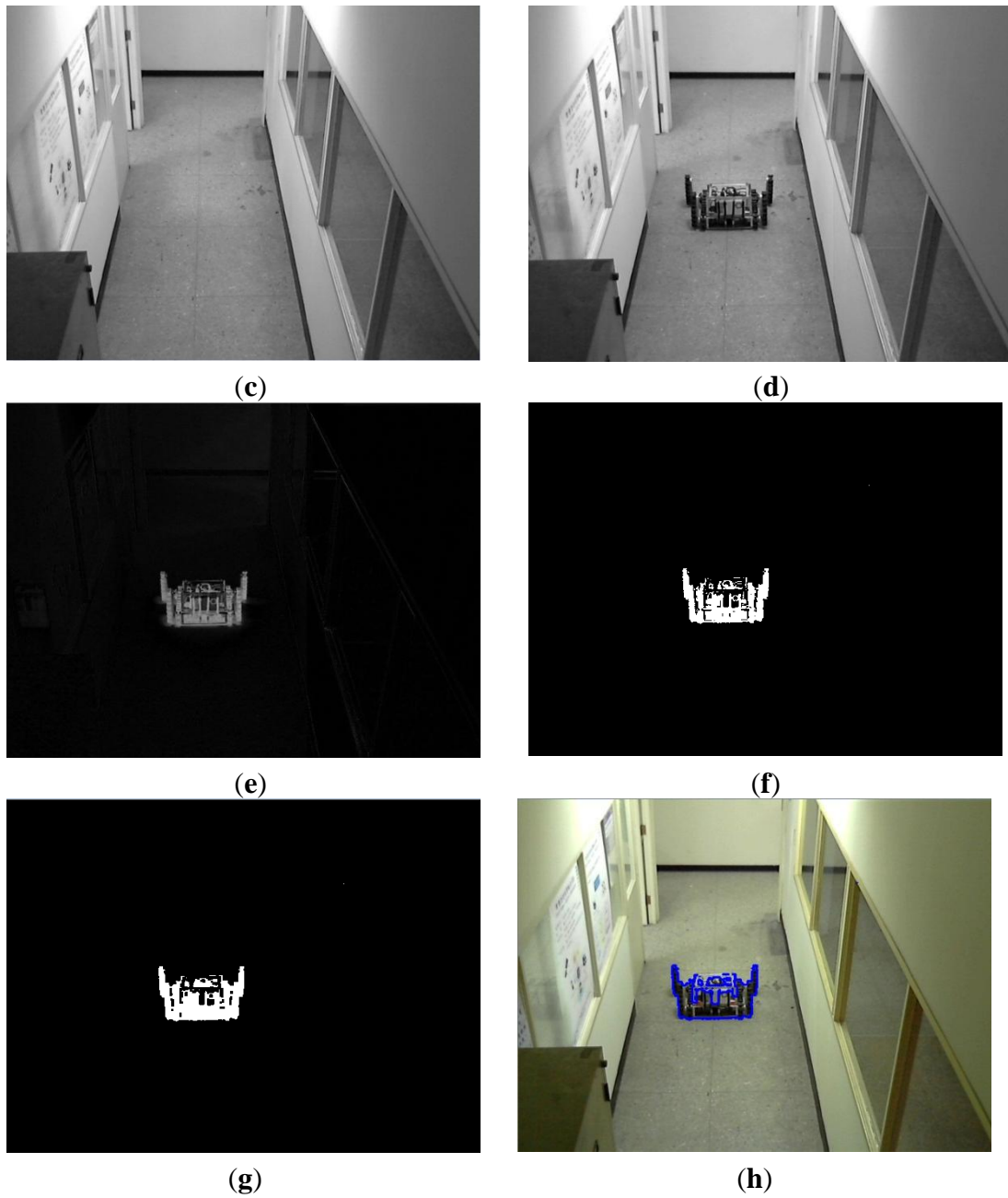
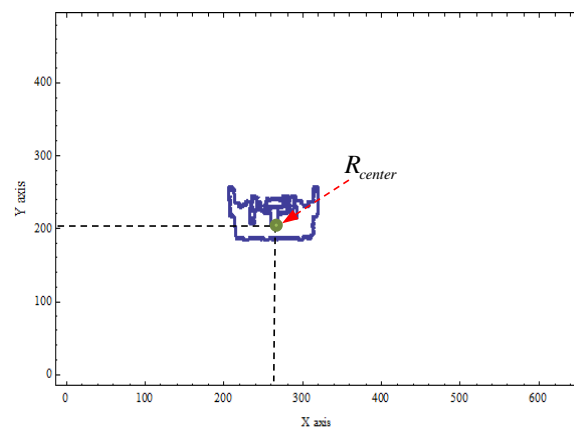


Figure 4. Coordinates of the mobile robot in the image-domain.





### 3. Mobile Robot Localization System with Single Webcam

After the captured images go through image processing, we can locate the robot in the image-domain. Then, we should calculate the coordinates of the robot in the image. That is, two distances, the x-axis and the y-axis, should be determined: 1.  $d_i$  represents the distance between  $R_{center}$  and the webcam: 2.  $w_i$  represents the distance between  $R_{center}$  and the wall, as shown in Figure 4. In this paper the IBDMS is used to calculate the distance  $d_i$ , and the PLDMS is used to calculate the distance  $w_i$ .

#### 3.1. Experimental Map

Figure 5 shows the map of our experimental environment. In the map, the coordinate of the first webcam is set to  $(x_1, y_1)$ , the second one is set to  $(x_2, y_2)$ , and the third one is set to  $(x_3, y_3)$ .  $wall_i$  ( $i=1,2,3$ ) represents the distance between the  $i$ th webcam and the wall.  $d_i$  ( $i=1,2,3$ ) represents the distance between the  $i$ th webcam and  $R_{center}$ .  $w_i$  ( $i=1,2,3$ ) is the distance between the wall and  $R_{center}$  in the  $i$ th webcam covering area. Clearly, the coordinates of webcams  $(x_i, y_i)$  and the distances  $wall_i$  are given. Therefore, if distances  $d_i$  and  $w_i$  can be calculated, we can easily represent the robot with its coordinate in the area, which is covered by one of the set-up webcams. The equations for calculating the coordinate of the robot are defined in Equations (15–17):

$$webcama_1 : \begin{cases} X_1 = x_1 + d_1 \\ Y_1 = y_1 + (wall_1 - w_1) \end{cases} \quad (15)$$

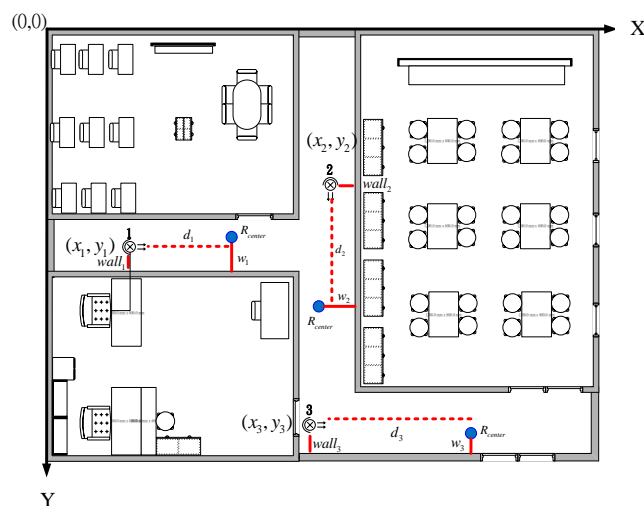
$$webcama_2 : \begin{cases} X_2 = x_2 - (wall_2 - w_2) \\ Y_2 = y_2 + d_2 \end{cases} \quad (16)$$

and:

$$webcama_3 : \begin{cases} X_3 = x_3 + d_3 \\ Y_3 = y_3 + (wall_3 - w_3) \end{cases} \quad (17)$$

where  $(X_i, Y_i)$  ( $i=1,2,3$ ) are the coordinate of the robot in the covering area of the  $i$ th webcam.

**Figure 5.** Map of our experimental location.



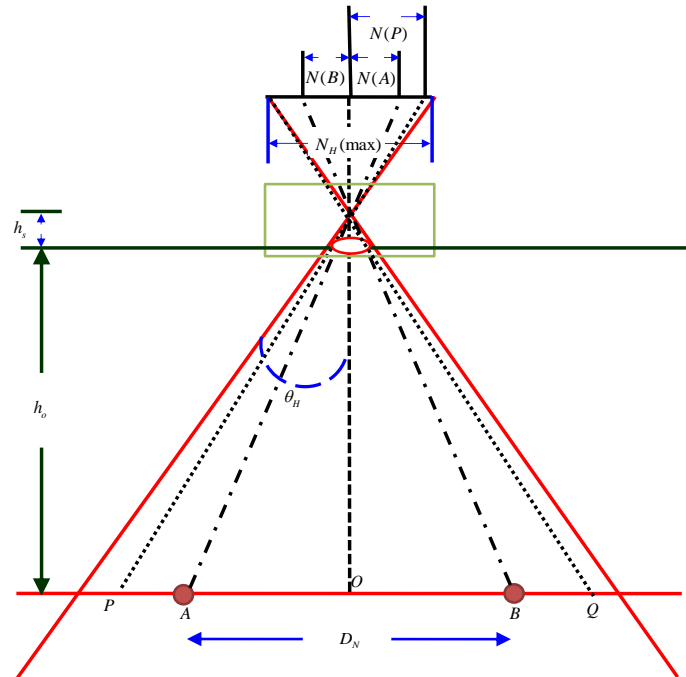
### 3.2. Calculation of Distance $d_i$ with IBDMS [24–29]

IBDMS is developed in this paper for the purpose of calculating the distances  $d_i$  ( $i = 1, 2, 3$ ), which can work on a single webcam and only depends on a known-dimension rectangle, *i.e.*, a ground tile. The idea of IBDMS is from the triangular relationship, shown in Figure 6, that is we first capture an image incorporating a known-dimension rectangle, and then the proportion relationship between the real-dimension and the image-dimension of the rectangle can be found. According to the proportion relationship, the distance  $d_i$  can then be easily calculated. Figure 6 shows the IBDMS set-up. It only requires a webcam and two given-location points  $A$  and  $B$ , which could be two corners of a ground tile.  $h_s$  is a constant parameter of the webcam.  $O$  is the intersection of the optical axis and the plane. The targeted objects which lie on the plane and are perpendicular to the optical axis can be measured by simple trigonometric function derivations. Hence,  $h_o$  can be expressed as:

$$h_o = \frac{D_N}{N(A) + N(B)} N_{H \max} \cot \theta_H - h_s \quad (18)$$

where  $h_o$  can be considered as any of distances  $d_i$  ( $i = 1, 2, 3$ ) for the  $i$ th webcam,  $N(A)$  and  $N(B)$  are pixel values in the captured image.  $N_H(\max)$  is the maximal pixel width in a horizontal scan line of the image.  $D_N$  is the width between points  $A$  and  $B$ .  $\theta_H$  is the horizontal view angle.

**Figure 6.** Schematic diagram of the IBDMS.



### 3.3. Calculation of Distance $w_i$ with PLDMS

Inheriting the concept of the IBDMS, a parallel-line distance measurement system (PLDMS) for measuring the distances  $w_i$  is developed. Figure 7 shows the schematic diagram of the PLDMS. In PLDMS, four points,  $LS_1^i$ ,  $LS_2^i$ ,  $LS_3^i$ , and  $LS_4^i$ , are considered as the reference points. In Figure 7, we draw a pair of parallel lines ( $LS_1^i LS_2^i$  and  $LS_3^i LS_4^i$ ) through these points.  $D_N$  is the width between

$\overline{Ls_1^i Ls_2^i}$  and  $\overline{Ls_3^i Ls_4^i}$ . Furthermore, in the image-domain, as Figure 7, the linear proportion of the line ( $\overline{Ls_3^i Ls_4^i}$ ) and the line ( $\overline{Ls_1^i Ls_2^i}$ ) can be defined as:

$$\overline{Ls_1^i Ls_2^i} : \frac{y^i - y_1^i}{y_2^i - y_1^i} = \frac{x^i - x_1^i}{x_2^i - x_1^i} \tag{19}$$

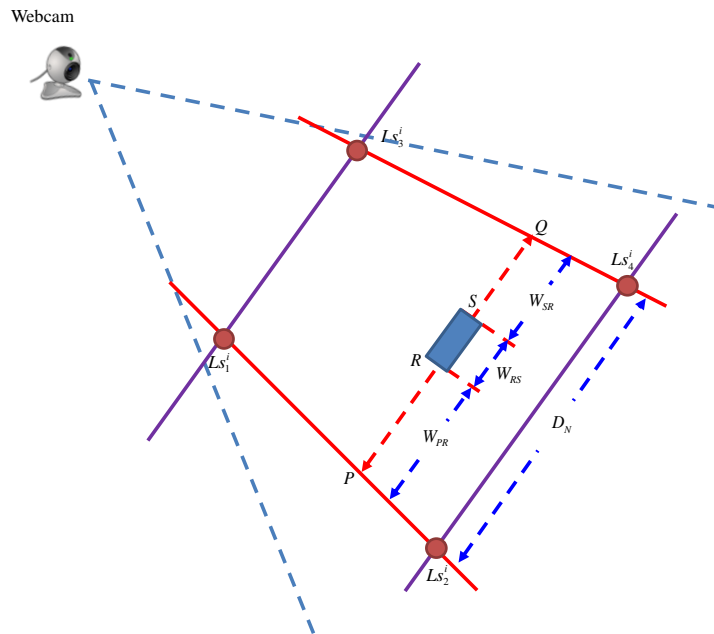
$$\overline{Ls_3^i Ls_4^i} : \frac{y^i - y_3^i}{y_4^i - y_3^i} = \frac{x^i - x_3^i}{x_4^i - x_3^i} \tag{20}$$

and:

$$N_H(PQ) = \overline{Ls_1^i Ls_2^i}(y^i)_{l_{PQ}} - \overline{Ls_3^i Ls_4^i}(y^i)_{l_{PQ}} \tag{21}$$

where  $(x_j^i, y_j^i)$  ( $j=1,2,3,4$ ) are, respectively, the image coordinate of the points  $Ls_j^i$ , and the pixel  $(x^i, y^i)$  ( $i=1,2,3$ ) is any image points laying at the line  $\overline{Ls_1^i Ls_2^i}$  or  $\overline{Ls_3^i Ls_4^i}$ . In Figure 7,  $N_H(PQ)$  is the number of pixels between  $P$  and  $Q$ , which can be obtained by Equation (21).

Figure 7. Diagram of the PLDMS.

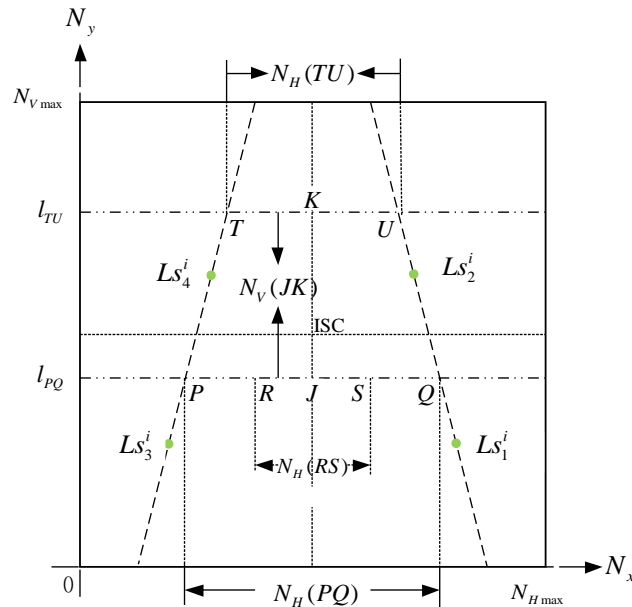


In Figure 8, points  $R$  and  $S$  could be two corners of a known-dimension rectangle, *i.e.*, a ground tile, in a captured image (image-domain), and  $P$  and  $Q$  are the cross points between the scan line  $l_{PQ}$  and the lines  $\overline{Ls_1^i Ls_2^i}$  and  $\overline{Ls_3^i Ls_4^i}$ . For the reason that  $P$ ,  $Q$ ,  $R$ , and  $S$  are laying at the same scan line  $l_{PQ}$ , we can easily calculate the points  $P$  and  $Q$  by Equations (19) and (20), where  $R, S$  and  $(x_j^i, y_j^i)$  ( $j=1,2,3,4$ ) have been given by the image processes. In our experiments, the width between  $R$  and  $S$  means the width of the robot. Furthermore, the width  $W_{RS}$  can be calculated by the proportion relationship, which is defined as:

$$W_{RS} = D_N \times \frac{N_H(RS)}{N_H(PQ)} \tag{22}$$

where  $N_H(RS)$  and  $N_H(PQ)$  can be obtained by the PLDMS. With the same idea, we can calculate the width  $W_{PR}$  or the width  $W_{SQ}$ .

Figure 8. Schematic diagram of the PLDMS.



3.4. Overall Procedures of the Proposed Localization Method

Figure 9 shows the overall scheme of the proposed robot localization method, where  $t_s$  is the update time-interval for the background image. The moment of updating the background image is that if the segmentation horizontal projection of the binary foreground image  $f_{fg}(x, y)$  is bigger than the threshold value  $t_1$ , the captured image  $f_{current}(x, y)$  is set to a new background image; otherwise, we keep the original background image. In Equation (23), the function  $s(\bullet)$  means the process of the segmentation horizontal projection, as shown in Figure 10.

$$f_{bg}(x, y) = \begin{cases} f_{current} & \text{if } s(f_{fg}(x, y)) < t_1 \\ f_{current}(x, y) & \text{otherwise,} \end{cases} \quad (23)$$

In the image processing, the coordinates of the robot can be calculated through the methods of gray-scale transformation, background subtraction, binarization, morphological processing, connected components labeling, and averaging method. The background image is updated in every  $t_s$  seconds with the update Equation (23). After image processing, the coordinate in the image-domain goes through the IBDMS for calculating the distance  $d_s$  (Equation (18)) and the PLDMS for calculating the width  $w_i$  (Equation (22)). Then, we can locate the robot in the real-world domain.

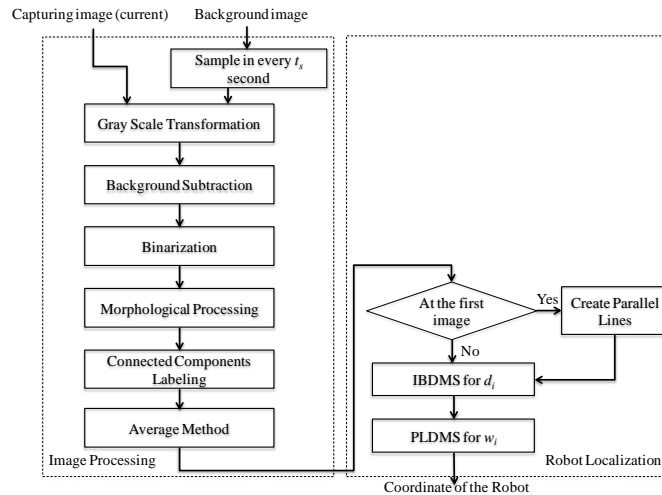
4. Experimental Results

4.1. Set-Up of the Experimental Environment

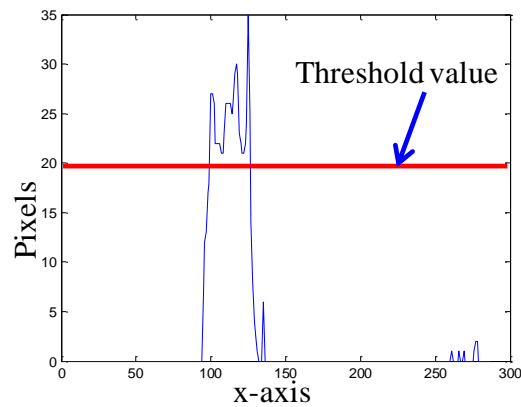
Figure 11 shows the map of our experimental environment. In this map three webcams, which are explained in Section 2.1, are used to cover the possible working area of the robot. The coordinates  $(x_1; y_1)$  of the first webcam are (279; 570);  $(x_2; y_2)$  are (755; 465); and  $(x_3; y_3)$  are (650; 1,030). Because of limitation of the USB 2.0 transmission speed, the resolution of the first webcam is  $640 \times 480$ ; the second webcam is  $640 \times 480$ ; and the third webcam is  $480 \times 360$ . The aisle widths are shown in

Figure 11, and are 138, 150 and 152 cm, respectively. The background image is updated in every 5 s. Threshold value  $t_1$  of the updated background image is set to 19.5.

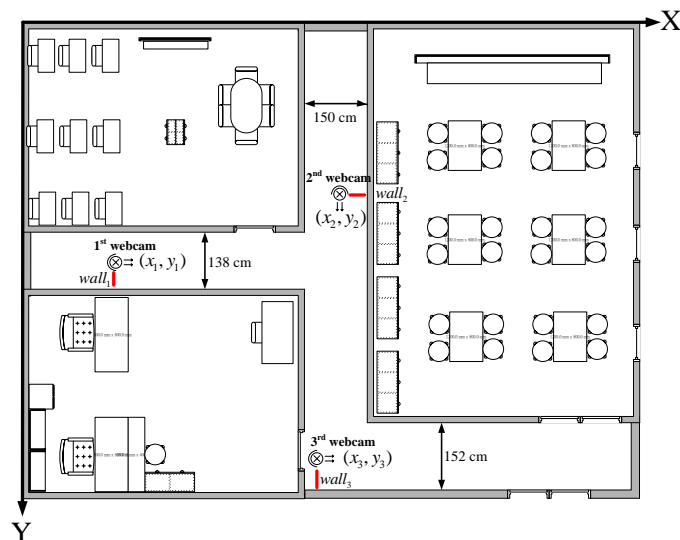
**Figure 9.** Overall scheme of the proposed localization method.



**Figure 10.** Update of the background image by using segmentation horizontal projection.



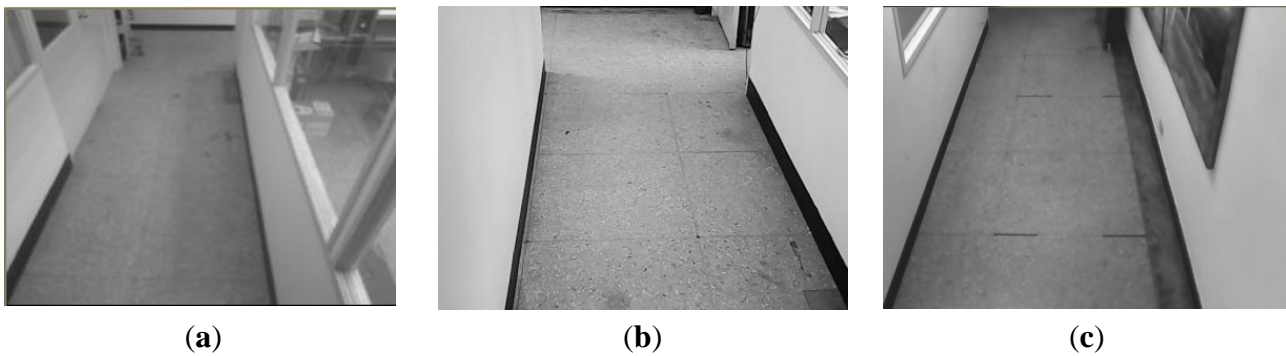
**Figure 11.** Map of our experimental environment.



#### 4.2. IBDMS and PLDMS Set-Up

Some basic set-up steps must be performed for the IBDMS and PLDMS before we start the robot localization procedures. In order to alleviate the impact from the image noise, building the first background image is done taking the average of 150 consecutive images, and then the background subtraction method can much more effectively extract the foreground image. Besides, a low-pass filter is adopted to further refine the background image. The obtained background images are shown in Figure 12.

**Figure 12.** (a) Background image of the 1st webcam. (b) Background image of the 2nd webcam. (c) Background image of the 3rd webcam.



In the 1st webcam, as shown in Figures 13 and 14, four corners of the known-dimension ground tile are used to draw a pair of the virtual parallel line ( $\overline{Ls_1^1 Ls_2^1}$  and  $\overline{Ls_3^1 Ls_4^1}$ ), whose deriving linear equations can be expressed as:

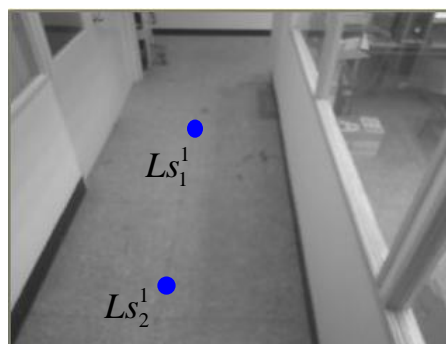
$$\overline{Ls_1^1 Ls_2^1} : y_1^1 = -6.0625x_1^1 + 865.063 \quad (24)$$

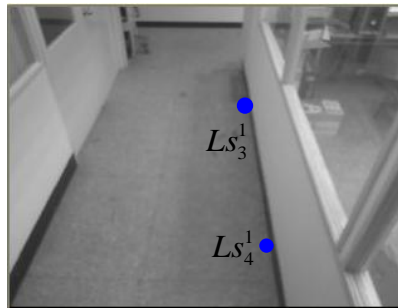
and:

$$\overline{Ls_3^1 Ls_4^1} : y_2^1 = 5.68x_2^1 - 992 \quad (25)$$

where the coordinates of  $Ls_1^1$ ,  $Ls_2^1$ ,  $Ls_3^1$ , and  $Ls_4^1$  are respectively chosen as (113, 180), (129, 83), (112, 181), and (128, 97) in the image-domain.

**Figure 13.** Two chosen points for the right line in the 1st webcam.



**Figure 14.** Two chosen points for the left line in the 1st webcam.

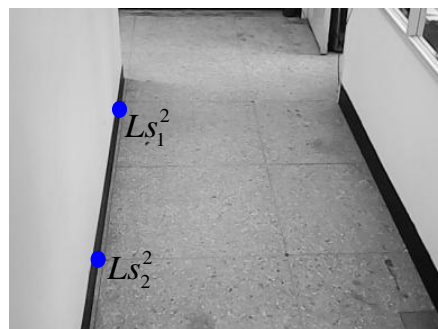
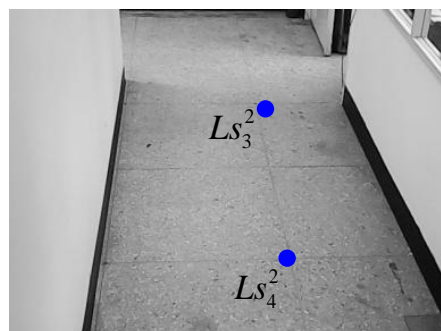
Similar to the setting procedures of the 1st webcam, the virtual parallel line ( $\overline{Ls_1^2 Ls_2^2}$  and  $\overline{Ls_3^2 Ls_4^2}$ ) for the 2nd webcam can be expressed as:

$$\overline{Ls_1^2 Ls_2^2} : y_1^2 = -8.107x_1^2 + 1500 \quad (26)$$

and:

$$\overline{Ls_3^2 Ls_4^2} : y_2^2 = 4.551x_2^2 - 1479.61 \quad (27)$$

where the coordinates of  $Ls_1^2$ ,  $Ls_2^2$ ,  $Ls_3^2$ , and  $Ls_4^2$  are respectively chosen as (140, 365), (168, 138), (404, 359), and (355, 136) in the image-domain. Figures 15 and 16 show these four points.

**Figure 15.** Two chosen points for the right line in the 2nd webcam.**Figure 16.** Two chosen points for the left line in the 2nd webcam.

The virtual parallel line ( $\overline{Ls_1^3 Ls_2^3}$  and  $\overline{Ls_3^3 Ls_4^3}$ ) for the 3rd webcam can be expressed as:

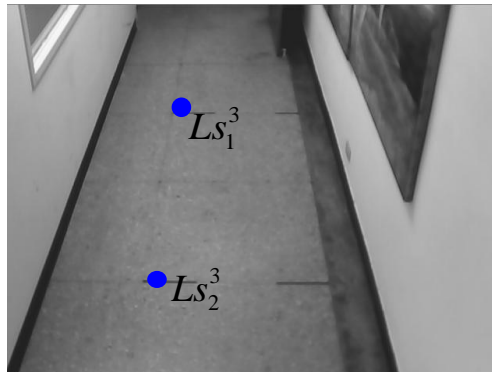
$$\overline{Ls_1^3 Ls_2^3} : y_1^3 = -5.475x_1^3 + 1309.65 \quad (28)$$

and:

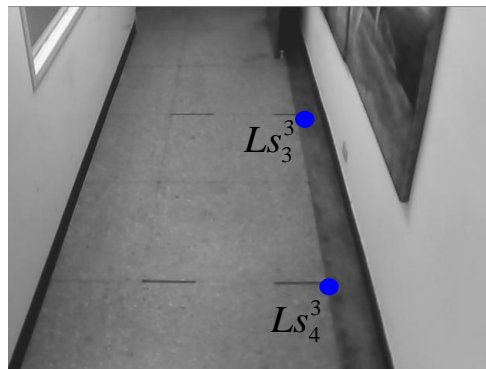
$$\overline{Ls_3^3 Ls_4^3} : y_2^3 = 5.815x_2^3 - 2055.55 \quad (29)$$

where the coordinates of  $Ls_1^3$ ,  $Ls_2^3$ ,  $Ls_3^3$ , and  $Ls_4^3$  are respectively chosen as (174, 357), (214, 138), (415, 358), and (377, 137) in the image-domain. Figures 17 and 18 show these four points.

**Figure 17.** Two chosen points for the right line in the 3rd webcam.



**Figure 18.** Two chosen points for the left line in the 3rd webcam.



### 4.3. Experimental Results

In our experiments, a remote-controlled track-robot moves through the monitored areas. The path of the robot is shown in Figure 19a. In Figure 19a, the circled locations, causing bigger errors as shown in Figure 19b, are the coordinates of the front arms of the robot at the moment of which the robot is moving into the covering area of the 2nd webcam, as shown in Figure 19c. In Figure 19d, a bigger error happens in the circled locations, which are the coordinates of the front arms of the mobile robot at the moment of which the robot is moving into the covering area of the 3rd webcam, as shown in Figure 19e. The error function used to show the ability of our proposed localization method is:

$$Error = \left| \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \right| \quad (30)$$

where  $(x_1, y_1)$  are the actual coordinates, and  $(x_2, y_2)$  are the coordinates measured by the proposed method. In Table 1, the measurement errors range from 2.24 cm (when the robot is near the webcams) to 12.37 cm (when the robot is far away from the webcam). According to the definition of  $R_{center}$  and the dimensions of the robot, which are 54 cm  $\times$  54 cm, a common size for patrolling robots, we find



that the robot can be correctly located even though it is far away from the webcams. Limited by the low-resolution webcams, the measurement errors are acceptable. We also can easily reduce the measurement errors by using high-resolution CCD camera. Furthermore, the selection of a known-dimension rectangle pattern should be clearly seen in the captured image in order to set up the reference points of IBDMS and PLDMS. In addition, a suitable threshold value  $t_1$  in the segmentation horizontal projection, which is used to update the background image, and the distortion coefficient vectors and camera conversion matrix in image calibration are important factors for precisely locating the mobile robots.

Under the conditions of the static monitored area, it is assumed that light sources and locations of walls and furniture are given. Light influence, therefore, can be easily attenuated through choosing appropriate factors in the image processing techniques. In this paper, we pay more attention to locating the moving robot by using single webcam and have not yet considered the situation of partial occlusions. In this static indoor environment, some image techniques [33–35] could be used to overcome temporary partial occlusion.

**Figure 19.** Moving path of the track robot (a) Path of the robot; (b) Path captured under the Webcam II (c) Image of the Webcam II; (d) Path captured under the Webcam III; (e) Image of the Webcam III

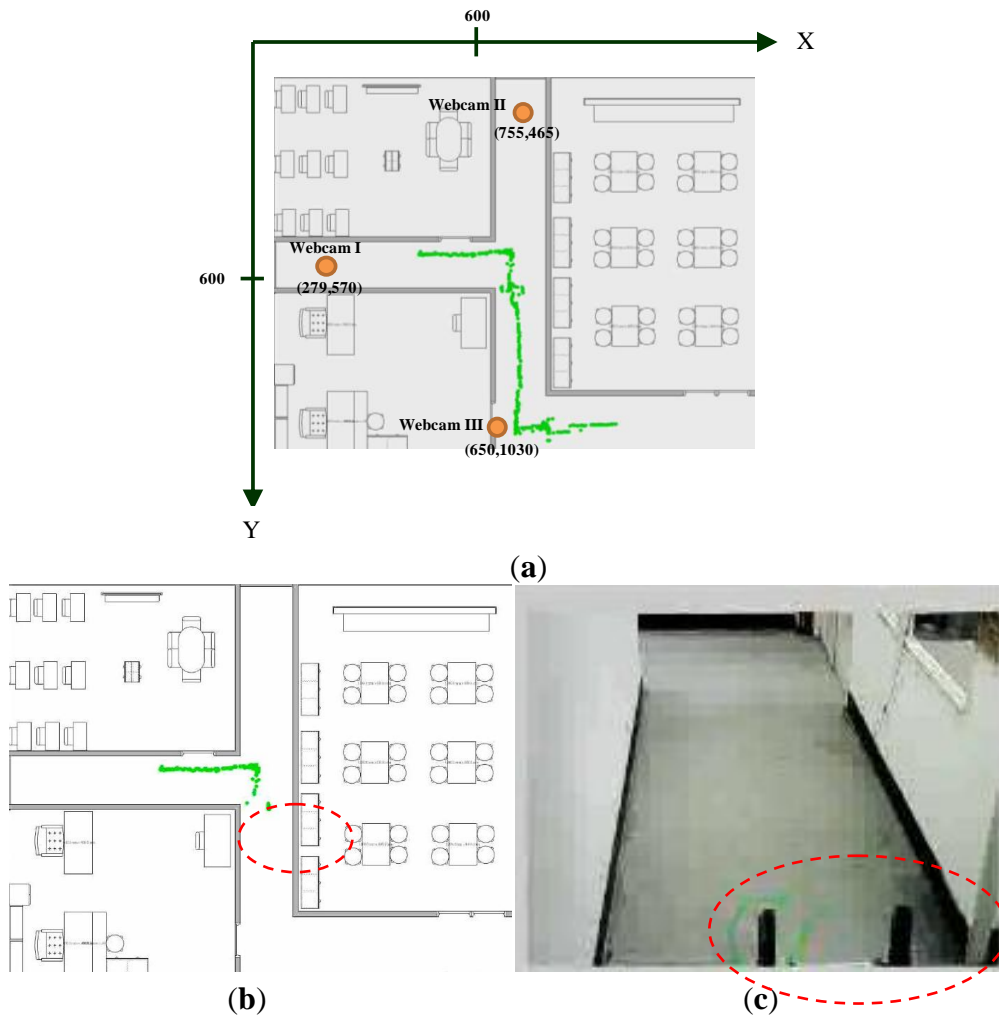


Figure 19. Cont.

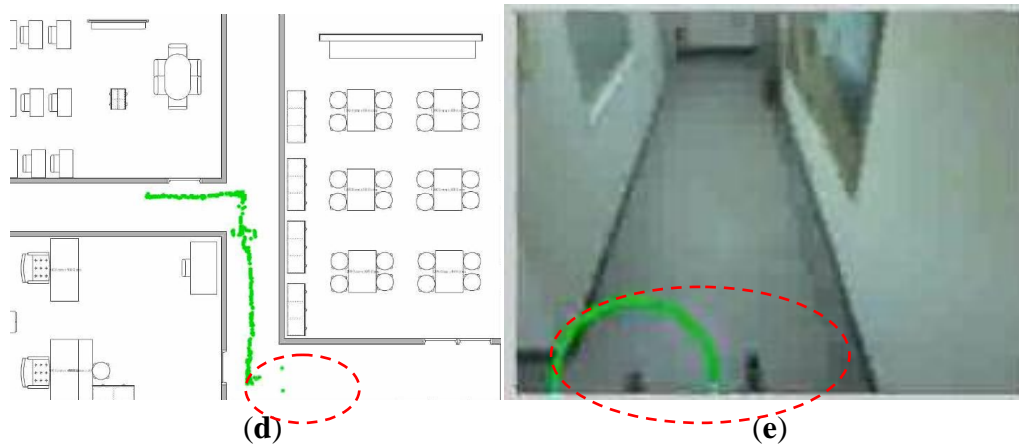


Table 1. Measurement errors of the proposed method.

Actual Coordinate	Measured Coordinate	Error (cm)	Webcams
(610,592)	(613,588)	5.00	Webcam I
(627,611)	(626,609)	2.24	Webcam I
(636,560)	(630,559)	8.08	Webcam I
(648,591)	(641,594)	7.62	Webcam I
(684,584)	(676,578)	10.00	Webcam I
(736,891)	(738,883)	8.24	Webcam II
(774,912)	(775,901)	11.04	<b>Webcam II</b>
(759,918)	(756,906)	12.37	Webcam II
(763,944)	(759,932)	12.65	Webcam II
(808,966)	(811,952)	12.37	Webcam II
(1001,779)	(997,779)	4.00	Webcam III
(1236,1076)	(1225,1074)	11.18	Webcam III
(1250,1019)	(1241,1021)	9.49	Webcam III
(1271,1063)	(1262,1065)	11.18	Webcam III
(1250,1041)	(1241,1039)	9.22	Webcam III

## 5. Conclusions

This paper proposes the use of IBMDS and PLDMS to locate a mobile robot in an indoor environment. Through the image processing and according to a known-dimension ground tile, the IBMDS and PLDMS used can calculate the coordinates of a moving tracked robot. Using this framework, we can quickly estimate the localization of the tracked robot. Furthermore, the experimental environment is easy to set up since only three parameters have to be defined, that is, the maximum pixel, the perspective, and the optical distance. Because the locations of webcams are fixed, we can utilize a simple background subtraction method to extract the data to attenuate the problem of computational burden. In addition, we use a low-pass filter and an on-line background updating method to reduce background noise, and we adopt the image morphology to acquire the robot's image information. This method does not use expensive high-resolution webcams and complex pattern

recognition methods to identify the mobile robot, but rather just uses a simple formula to estimate distance. From the experimental results, the localization method is both reliable and effective.

### Acknowledgment

This work was supported by the National Science Council, Taiwan, under Grants NSC 102-2221-E-003-009, NSC 102-2221-E-234-001-, and NSC 102-2811-E-011-019.

### Author Contributions

The work presented here was carried out in collaboration between all authors. I.-H. L. and M.-C. C. defined the research theme. I.-H. L. and W.-Y. W. designed methods and experiments, carried out the laboratory experiments, analyzed the data, interpreted the results and wrote the paper. S.-F. S. co-worked on associated data collection and their interpretation. T.-W. L. co-designed experiments, discussed analyses, interpretation, and presentation. All authors have contributed to, seen and approved the manuscript.

### Conflicts of Interest

The authors declare no conflict of interest.

### References

1. Kim, J.B.; Jun, H.S. Vision-based location positioning using augmented reality for indoor navigation. *IEEE Trans. Consum. Electr.* **2008**, *54*, 954–962.
2. Lin, C.H.; King, C.T. Sensor-deployment strategies for indoor robot navigation. *IEEE Trans. Syst. Man Cybernet.* **2010**, *40*, 388–398.
3. Hinckley, K.; Sinclair, M. Touch-Sensing Input Devices. In Proceedings of the Conference on Human Factors in Computing Systems, Pittsburgh, PA, USA, 15–20 May 1999.
4. Partridge, K.; Arnstein, L.; Borriello, G.; Whitted, T. Fast Intrabody Signaling. Demonstration at Wireless and Mobile Computer Systems and Applications, Monterey, CA, USA, 7–8 December 2000.
5. Hills, A. Wireless android. *IEEE Spectr.* **1999**, *36*, 49–53.
6. Want, R.; Hopper, A.; Falcao, V.; Gibbons, J. The active badge location system. *ACM Trans. Inform. Syst.* **1992**, *10*, 91–102.
7. Bahl, P.; Padmanabhan, V. RADAR: An In-Building RF Based User Location and Tracking System. In Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv, Israel, 26–30 March 2000.
8. Starner, T.; Schiele, B.; Pentland, A. Visual Context Awareness via Wearable Computing. In Proceedings of the International Symposium on Wearable Computers, Pittsburgh, PA, USA, 19–20 October 1998; pp. 50–57.
9. Ni, L.M.; Lau, Y.C.; Patil, A.P. LANDMARC: Indoor location sensing using active RFID. *Wirel. Netw.* **2004**, *10*, 701–710.

10. Subramanian, V.; Burks, T.F.; Arroyo, A.A. Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. *Comput. Electr. Agric.* **2006**, *53*, 130–143.
11. Barawid, O.C., Jr.; Mizushima, A.; Ishii, K. Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. *Biosyst. Eng.* **2007**, *96*, 139–149.
12. Sato, T.; Sugimoto, M.; Hashizume, H. An extension method of phase accordance method for accurate ultrasonic localization of moving node. *IEICE Trans. Fundament. Electr. Commun. Comput. Sci.* **2010**, *J92-A*, 953–963.
13. Carmer, D.C.; Peterson, L.M. Laser radar in robotics. *Proc. IEEE* **1996**, *84*, 299–320.
14. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, UK, 2005.
15. Hightower, J.; Want, R.; Borriello, G. *SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*; UW-CSE00-02-02, University of Washington: Seattle, WA, USA, 2000.
16. Lin, T.H.; Chu, H.H.; You, C.W. Energy-efficient boundary detection for RF-based localization system. *IEEE Trans. Mob. Comput.* **2009**, *8*, 29–40.
17. Lorincz, K.; Welsh, M. MoteTrack: A robust, decentralized approach to RF-based location tracking. *Pers. Ubiqu. Comput.* **2007**, *11*, 489–503.
18. Dellaert, F.; Fox, D.; Burgard, W.; Thrun, S. Monte Carlo Localization for Mobile Robots. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999; pp. 1322–1328.
19. Thrun, S.; Bücken, A.; Burgard, W.; Fox, D.; Frölinghaus, T.; Henning, D.; Hofmann, T.; Krell, M.; Schmidt, T. *Map Learning and High-Speed Navigation in RHINO*; MIT Press: Cambridge, MA, USA, 1998.
20. Li, S.; Liu, B.; Chen, B.; Lou, Y. Lou neural network based mobile phone localization using bluetooth connectivity. *Neural Comput. Appl.* **2013**, *23*, 667–675.
21. Li, S.; Qin, F. A dynamic neural network approach for solving nonlinear inequalities defined on a graph and its application to distributed. *Neurocomputing* **2013**, *117*, 72–80.
22. Martins, M.H.T.; Chen, H.; Sezaki, K. Otmcl: Orientation Tracking-Based Monte Carlo Localization for Mobile Sensor Networks. In Proceedings of the Sixth International Conference on Networked Sensing Systems, Pittsburgh, PA, USA, 17–19 June 2009; pp. 1–8.
23. Cheok, A.D.; Yue, L. A novel light-sensor-based information transmission system for indoor positioning and navigation. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 290–299.
24. Hsu, C.C.; Lu, M.C.; Wang, W.Y.; Lu, Y.Y. Three dimensional measurement of distant objects based on laser-projected CCD images. *IET Sci. Meas. Technol.* **2009**, *3*, 197–207.
25. Mertzios, B.G.; Tsirikolias, I.S. Applications of Coordinate Logic Filters in Image Analysis and Pattern Recognition. In Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, Pula, Croatia, 19–21 June 2001; pp. 125–130.
26. Lu, M.C.; Wang, W.Y.; Chu, C.Y. Image-based distance and area measuring systems. *IEEE Sens. J.* **2006**, *6*, 495–503.
27. Wang, W.Y.; Lu, M.C.; Kao, H.L.; Chu, C.Y. Nighttime Vehicle Distance Measuring Systems. *IEEE Trans. Circ. Syst. II: Exp. Brief.* **2007**, *54*, 81–85.

28. Wang, W.Y.; Lu, M.C.; Wang, T.H.; Lu, Y.Y. Electro-optical distance and area measuring method. *WSEAS Trans. Syst.* **2007**, *6*, 914–919.
29. Hus, C.C.; Lu, M.C.; Wang, W.Y.; Lu, Y.Y. Distance measurement based in pixel variation of CCD image. *ISA Trans.* **2009**, *40*, 389–395.
30. Osugi, K.; Miyauchi, K.; Furui, N.; Miyakoshi, H. Development of the scanning laser radar for ACC system. *JSAE Rev.* **1999**, *20*, 549–554.
31. Nakahira, K.; Kodama, T.; Morita, S.; Okuma, S. Distance measurement by an ultrasonic system based on a digital polarity correlator. *IEEE Trans. Instrum. Meas.* **2001**, *50*, 1478–1752.
32. Kawaji, H.; Hatada, K.; Yamasaki, T.; Aizawa, K. An Image-Based Indoor Positioning for Digital Museum Applications. In Proceedings of the 16th International Conference on Virtual Systems and Multimedia (VSMM), Seoul, Korea, 20–23 October 2010.
33. Cucchiara, R.; Grana, C.; Tardini, G. Probabilistic People Tracking for Occlusion Handling. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004; pp. 132–135.
34. Eng, H.L.; Wang, J.X.; Kam, A.H. A Bayesian Framework for Robust Human Detection and Occlusion Handling Human Shape Model. In Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004, pp. 257–260.
35. Tsai, C.Y.; Song, K.T. Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness. *Image Vis. Comput.* **2007**, *27*, 1007–1022.

© 2014 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/>).