Indoor Intelligent Mobile Robot Localization Using Fuzzy Compensation and Kalman Filter to Fuse the Data of Gyroscope and Magnetometer

Hung-Yuan Chung, *Senior Member, IEEE*, Chun-Cheng Hou, and Yu-Shan Chen

***Abstract*—This paper develops an indoor intelligent service mobile robot that has multiple functions, can recognize and grip the target object, avoid obstacles, and accurately localize via relative position. The locating method of the robot uses the output values of the sensor module, which includes data from a gyroscope and a magnetometer, to correct the current rotation direction angle of the robot. An angle correction method can be divided into three parts. The first part calculates the angle values obtained from the gyroscope and the magnetometer that are installed on the robot. The second part explores the error characteristics between the sensor module and the actual rotation direction angle of the robot. The third part uses the error characteristic data to design the fuzzy rule base and the Kalman filter to eliminate errors and to get a more accurate orientation angle. These errors can be described as either regular or irregular. The former can be eliminated by fuzzy algorithm compensation, and the latter can be eliminated by the Kalman filter. The contribution of this paper is to propose an error correction method between the calculus rotation angle determined by the sensor and the actual rotation angle of the robot such that the three moving paths, i.e., specified, actual, and calculus paths, have more accurate approximation. The experimental results demonstrate that the combination of fuzzy compensation and the Kalman filter is an accurate correction method.**

***Index Terms*—Fuzzy compensation, gyroscope, Kalman filter, localization, magnetometer, mobile robot.**

Nomenclature

$p_{\rm sp}$	Specified position, destination of robot.
$\theta_{\rm sp}$	Specified absolute direction angle.
$\theta_{\rm rotate}$	Commanded rotation angle of the robot.
θ_G	Absolute azimuth of the gyroscope.
θ_M	Absolute direction angle of the magnetometer.
$p_{\rm now}$	Current position of the robot.
$\theta_{\rm now}$	Current direction of the robot.
l_b	Distance unit of each forward movement of the robot (let $l_b = 2$ cm).
$\theta_{\rm set}$	Difference value between the current $\theta_{\rm now}$ and $\theta_{\rm sp}$.
$\check{\theta}_{\rm sp}(n)$	Relative direction angles of the specified position.
$\theta_{\rm sp}(n)$	Absolute direction angles of the specified position.
$\theta_{\rm KG}$	Gyroscope correction value calculated by Fuzzy + Kalman.
$\theta_{\rm KM}$	Magnetometer correction value calculated by Fuzzy + Kalman.
$\theta_{\rm kal}$	Correction value calculated by Fuzzy + Kalman (integrated with gyroscope and magnetometer).
$g_{\rm fuzzy}$	Gyroscope error calculated by fuzzy compensation.
$m_{\rm fuzzy}$	Magnetometer error calculated by fuzzy compensation.
$\theta_{\rm Gf}$	Gyroscope value calculated by fuzzy compensation.
$\theta_{\rm Mf}$	Magnetometer value calculated by fuzzy compensation.

I. Introduction

In recent years, intelligent service mobile robots have been receiving considerable attention not only from research institutes but also from commercial automation industries. Therefore, a growing number of studies are now available to shed some light on intelligent service robots. With respect to the robotic home service industry, several functions need to be considered, such as cleaning, home security, and other domestic tasks. Currently, commercially available robotic vacuum cleaner products can automatically clean the floors [1], [2]. Another robot product has a surveillance system and hopping locomotion capabilities for home security [3]. A surveillance patrol robot proposed in [4] can actively track a moving object and inform the householder regarding domestic security issues. A specific auxiliary robot is able to detect and grasp typical household objects and to safely exchange them with humans [5], [6]. In the special-purpose realm, a robot manufactured by REMOTEC can carry out hazardous tasks [7]. An explosives-removing robot that can remove explosives to a safer area was proposed in [8]–[10]. Some studies even combine machine vision enhancement with the ability of object tracking [11]–[14]. In addition, other important and contributive research regarding intelligent robots has been proposed in [15]–[21].

This paper presents an indoor intelligent service mobile robot to help humans transport objects. To achieve this function, an iNEMO sensor module [22], a Kinect camera, a digital signal

Manuscript received August 22, 2014; revised January 29, 2015; accepted February 27, 2015. Date of publication March 26, 2015; date of current version September 9, 2015. This work was supported by the Ministry of Science and Technology (MOST) of Taiwan under Grant MOST 103-2218-E-008-007.

The authors are with the Department of Electrical Engineering, National Central University, Taoyuan 32001, Taiwan (e-mail: hychung@ee.ncu.edu.tw; 955401018@cc.ncu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2015.2416692

processing (DSP) chip, four infrared (IR) distance measurement circuits, and a mechanical arm are combined. Using the given combination, one can implement a localization system, an image recognition function, obstacle avoidance, and catching of a specific target object. The iNEMO sensor module is made by STMicroelectronics, and it has several highly accurate microelectromechanical system sensors. The Kinect camera is manufactured by Microsoft. The DSP chip (MS320F28335), which is manufactured by Texas Instruments, is a platform for communication between the IR sensors, the mechanical arm, and a laptop. The iRobot Create product is made by iRobot Corporation. The iRobot product has been modified to our desired functionality. It uses the Kinect camera to capture the image in front of the iRobot and several image-processing algorithms to recognize any target object in the input image. If the target object is present, the robot will be programmed to move toward the target to catch it. Simultaneously, the iNEMO module is used to realize the localization system and obtain the current position of the robot. The output of the sensors can be used to analyze the action and realize the localization of the robot.

The localization technology is also an important issue in mobile robots. It can be divided into two parts, i.e., the absolute and relative coordinates. The former uses sensors and a tag to mark a distance (whose positions are known) to calculate the absolute position of the robot, such as in [23] and [24], which use IR and ultrasonic sensors and radio-frequency identification tags [25]–[32] to detect robot movement. This paper examines the latter, i.e., the relative position. Each time the robot moves, the distance and the direction of movement of the robot are used to compute the new position. Three-dimensional coordinates (x, y, and θ) are used in the 2-D environment to construct a map, as in [33]–[35], using ultrasonic sensors [36], [37] image sensors, and other types of sensors. GPS cannot be employed in this case as our system is examined indoors. The amount of calculation and the technical requirements are much lower for relative positioning than that for absolute positioning. The Kalman filter [38] is often used in the optimization of the sensor output to eliminate irregular noise. The efficiency of the algorithm and the estimation precision are high. In this paper, the fuzzy compensation method is presented to describe the regular error characteristics of sensor output, and then the Kalman filter is used to eliminate the uncertainty in irregular errors. After obtaining the precise estimation value of the angle and the information regarding the distance the robot movement, the locating method can be executed. In implementing this experiment, the robot is programmed to move to a preset distance that is determined based on target object position. Moreover, the current position of the robot can be calculated by the locating method. The purpose of this paper is to approximate the specified, actual, and calculus paths to achieve accurate localization results. The experimental results show that combining fuzzy compensation and Kalman filter yields higher location accuracy than that by using fuzzy compensation alone.

This paper is organized as follows. Section II describes the hardware, system architecture, and the method of obtaining the gyroscope and magnetometer data. The angle correction and image recognition methods are described in Section III.

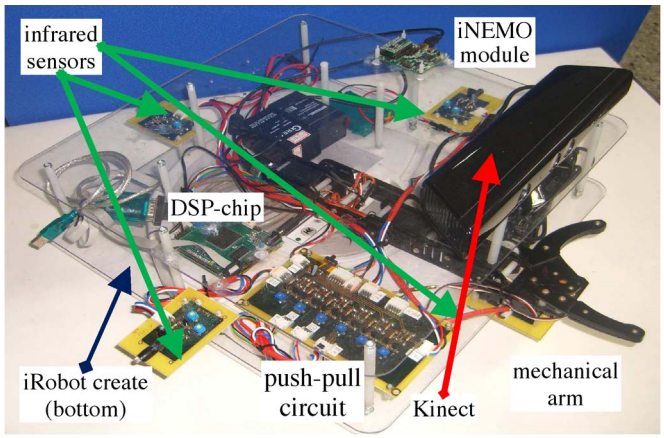

Fig. 1. Hardware structure.

The experiments and results are described in Section IV. The conclusion of this paper is described in Section V.

II. Hardware of Control System

A. System Architecture

The robot presented in this paper is shown in Fig. 1, and it comprises of a Kinect camera, a laptop, an iNEMO sensor module, a DSP chip, a mechanical arm, iRobot Create, and IR sensors.

1) Control System Architecture: The architecture of the control system is shown in Fig. 2. The gyroscope and magnetometer data are obtained from the iNEMO module. These data are used to locate the robot and to predetermine the position to place the target after it is grabbed. The target object image information is captured by the Kinect camera, which recognizes the image and controls the robot to track the target in real time. For arm grabbing and obstacle avoidance, the IR distance measurement circuits are installed on the front, back, left, and right sides of the robot to measure the distance between obstacles and the robot. The DSP chip provides the pulsewidth modulation (PWM) drive signal required by the IR circuit and processes the distance data from the IR receiver circuit; thereafter, it uses the RS232 communication interface to transmit data to a laptop. By combining the aforementioned multiple functions, our system can improve upon the existing traditional vacuuming robot that only cleans fine dust on the floor and avoid obstacles only after impact. The main objective is to add functionality to the home cleaning robots, such as cleaning the floor, and grabbing trash on the floor and moving it to a specified location. Moreover, our obstacle avoidance function enables the robot to avoid obstacles at a distance without the need to impact the object first. A laptop is used to integrate all functions of robot motion with Visual Studio 2010-Express as the software development interface. By integrating data from the Kinect camera, the IR distance measurement circuit, and the mechanical arm, our system achieves robot localization, target tracking, obstacle avoidance, and object delivery to a specified location.

2) Localization System Architecture: The localization system architecture is shown in Fig. 3. First, specified position p_{sp} and absolute direction angle θ_{sp} of the robot are imported

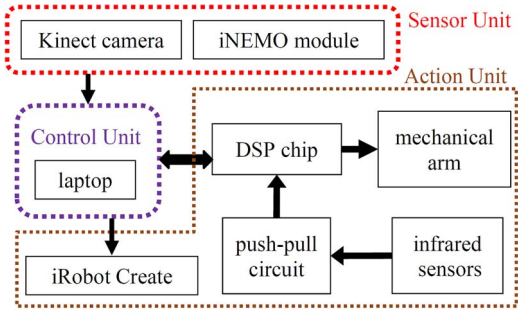

Fig. 2. Architecture of the control system.

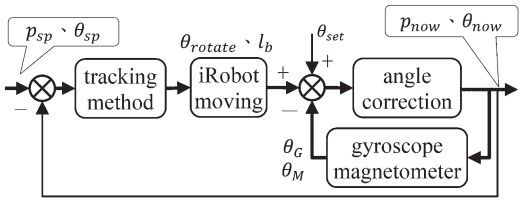

Fig. 3. Localization system block diagram.

into the control unit. The control command is issued to iRobot using the specified position tracking method. The control command is sent/received by the Bluetooth module of iRobot and a laptop, and the iRobot rotates θ_{rotate} and moves l_b. θ_{set} is the difference value between the current θ_{now} and the θ_{sp}. The robot rotates, and the control unit accesses the absolute azimuth data θ_G of the z-axis of gyroscope. The absolute direction angle θ_M of the x-axis and the y-axis of the magnetometer is accessed when the robot stops. The values are taken by the iNEMO module and transferred via the universal serial bus (USB) to the control unit. The control unit calculates the current absolute direction angle θ_{now} and position p_{now} of the robot by weighted statistics, fuzzy compensation, and Kalman filter, respectively. Afterward, the system continues recursive computation, and the robot moves toward the specified position.

B. iNEMO Module

This module contains a three-axis gyroscope and a three-axis magnetometer. In this paper, the gyroscope and the magnetometer are used to obtain the relative and absolute orientation angles of the robot, and then the fuzzy compensation and the Kalman filter are applied to achieve precise angular localization. The sensing module is placed on top of the robot to avoid the magnetic force from affecting the iRobot motor's magnetic field. The data of the angles can be obtained from the gyroscope and magnetometer within the iNEMO sensor module, wherein the gyroscope provides the relative angle of rotation of the robot and the magnetometer uses the measured strength of the Earth's magnetic field to infer the perspective of an absolute angle.

In this paper, the relative angle of the gyroscope is converted into an absolute angle by accumulation. By using the angular velocity of the z-axis of the gyroscope and taking the sampling time to integral, the numerical methods can be used to obtain approximations for the current value of the rotation angle of the robot g_{angle}; the calculation method is very simple, and the influence of calculation error is small. By adding the previous

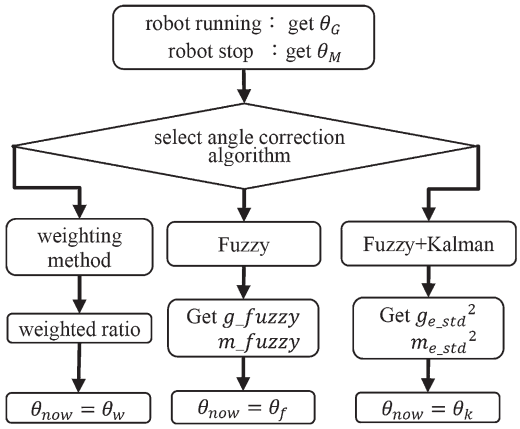

Fig. 4. Flowchart of angle correction.

absolute direction angle $\theta_{\text{now}}(n-1)$, this can obtain a current absolute angle of the gyroscope θ_G

$$g_{\text{angle}} = t\sum_{i=0}^{n} G_{z_i} \tag{1}$$

$$\theta_G(n) = \theta_{\text{now}}(n-1) + g_{\text{angle}} \tag{2}$$

where G_{z_i} is the angle of rotation of the gyroscope obtained every 0.04 s, i.e., $t = 0.04$ s, when the robot is rotating, and $\theta_G(n)$ is the nth absolute angle of the gyroscope.

Because this system is horizontal, the roll and pitch of the robot do not need to be considered, only the yaw (m_x and m_y), where m_x and m_y are the yaw data obtained from the magnetometer. Once the angle values of the gyroscope have been taken, the corrected absolute direction angle θ_M of the robot can then be obtained via the magnetometer when the robot stops, as follows:

$$\theta_M = \begin{cases} \tan^{-1}\left(\frac{m_y}{m_x}\right) & ;m_y > 0, m_x > 0 \\ 180 + \tan^{-1}\left(\frac{m_y}{m_x}\right) & ;m_y > 0, m_x \le 0 \\ 360 + \tan^{-1}\left(\frac{m_y}{m_x}\right) & ;m_y \le 0, m_x > 0 \\ 90 & ;m_y < 0, m_x = 0 \\ 270 & ;m_y > 0, m_x = 0. \end{cases} \tag{3}$$

III. Methods

A. Angle Correction Algorithms

The angle correction flowchart is shown in Fig. 4, where θ_w, θ_f, and θ_k are the direction angle of the robot after the weighted method, fuzzy compensation, and Kalman filter combined with fuzzy compensation, respectively. $g_$fuzzy and $m_$fuzzy are the error values of the gyroscope and magnetometer after defuzzification, respectively. $g^2_{e\_\text{std}}$ and $m^2_{e\_\text{std}}$ are the process error covariance and measurement error covariance of the Kalman algorithm, respectively (see Table I).

An angle correction operation is performed at the next rotation of the robot. Fig. 5 shows the explanation of the angle correction.

TABLE I
SYMBOL TABLE

Kalman Filter Symbols	Symbols in this study
$\hat{x}_{k\|k-1}$: predicted state estimate	θ_{KG}: gyroscope correction
$\hat{x}_{k-1\|k-1}$: best predicted state estimate	θ_G: absolute angle
$\hat{x}_{k\|k}$: updated state estimate	θ_{kal} : state estimate
$p_{k\|k-1}$: predicted estimate error covariance	$p_1(k)$
$p_{k-1\|k-1}$:best predicted estimate error covariance	$p(k-1)$
$p_{k\|k}$: updated estimate error covariance	$p(k)$
$M_{k\|k}$: measurement	θ_{KM} : magnetometer correction
K_g : optimal Kalman gain	k_gain
W_k : process error	g_{fuzzy}
V_k : measurement error	m_{fuzzy}
Q_k : process error covariance	${g_{e\_std}}^2$
R_k : measurement error covariance	${m_{e\_std}}^2$
A_k : state transition matrix	unit matrix $[1]_{1\times1}$
B_k : control-input matrix	
C_k : observation matrix	

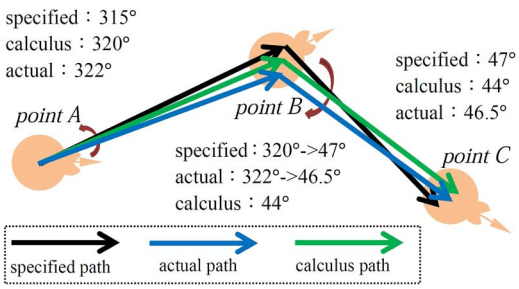

Fig. 5. Schematic angle correction.

From "*Point A*" to "*Point B*"— Assuming the specified angle is 315°, the control unit calculates the current angle of the robot as 320°, and their error is 5°. The actual angle measured artificially is 322°. The robot continues to travel forward, and the trajectory has a little error at this process. From "*Point B*" to "*Point C*" — In order to allow the robot to reach the target point C, it must rotate by 47° from *point B*. The specified angle from the correct original 315° → 47° (which means to turn right by 92°) is corrected to 320° → 47° (which means to turn right by 87°) to compensate for the degree error (5°) in the specified path after the calculation that had occurred at point *A*. The real amount at this angle is measured out from 322° → 46.5° (which means to turn right 85°); therefore, the process of moving from point B to point C can shorten the error of the specified and actual angles. Continue to assume that the current angle of calculating the robot's position is 44°. Although there is an angle error between the actual angle and the calculus angle that still occur in each correct processing, this error will be compensated after the robot moves to the next specified point (e.g., *point C*).

1) Statistical Weighting Method: This algorithm is weighted according to the sensors and the error of specified angle. Due to this approach, it does not completely describe the error distribution of the two sensors; therefore, it is not possible to predict the actual angle effectively. Instead, it was used as a contrast with other algorithms in this paper. Assume the weighted ratio of the gyroscope and the magnetometer is $\omega_1 : \omega_2$

$$\text{Then } \theta_{\text{now}}(n) = \theta_w(n) = \theta_G(n) \times \omega_1 + \theta_M(n) \times \omega_2. \quad (4)$$

2) Fuzzy Compensation: This algorithm is used to organize the robot's actual collected rotation angles, as well as the errors of the gyroscope or magnetometer, into a fuzzy formula, and then make the error correction.

a) Fuzzy rules of the gyroscope error: The gyroscope errors are organized into the following fuzzy rule base:

$$\begin{aligned} R_{g1}&: \text{If } 0^\circ \le g_{\text{angle}} \le \theta_1, && \text{Then } g_{e1} = k_1(G\_\text{value}) \\ &\vdots \\ R_{gr}&: \text{If } \theta_m \le g_{\text{angle}} \le 360^\circ, && \text{Then } g_{er} = k_r(G\_\text{value}) \end{aligned} \quad (5)$$

where r is the number of fuzzy rule base; $R_{g1} \sim R_{gr}$ are the fuzzy rules; $g_{e1} \sim g_{er}$ are the error between the gyroscope and the actual angle; gyroscope value (G_value) is measured while the robot is rotating; $k_1 \sim k_r$ are the functions of the G_value, as shown by the Trend_Line in Fig. 10; and $\theta_1, \ldots, \theta_m$ are the angles from 0° to 360°.

The weighted average method is used for defuzzification, and the gyroscope error g_{fuzzy} is obtained as follows:

$$g_{\text{fuzzy}} = \frac{\sum_{i=1}^{r} \alpha_i g_{ei}}{\sum_{i=1}^{r} \alpha_i} \quad (6)$$

where $i = 1, \ldots, r$. If g_{angle} meets a fuzzy rule, let the firing strength $\alpha_i = 1$; otherwise, let $\alpha_i = 0$.

Finally, the gyroscope value $\theta_{\text{Gf}}(n)$ is the sum of absolute azimuth θ_G and g_{fuzzy} of the gyroscope, i.e.,

$$\theta_{\text{Gf}}(n) = \theta_G(n) + g_{\text{fuzzy}}. \quad (7)$$

b) Fuzzy rules of the magnetometer error: The magnetometer errors are organized into the following fuzzy rule base:

$$\begin{aligned} R_{m1}&: \text{If } 0^\circ \le \theta_{\text{set}} \le \theta_1, && \text{Then } m_{e1} = k_1(M\_\text{value}) \\ &\vdots \\ R_{mr}&: \text{If } \theta_m \le \theta_{\text{set}} \le 360^\circ, && \text{Then } m_{en} = k_r(M\_\text{value}) \end{aligned} \quad (8)$$

where $R_{m1} \sim R_{mr}$ are fuzzy rules, and $m_{e1} \sim m_{er}$ are the error between the magnetometer and the actual angle. M_value is the magnetometer value, $k_1 \sim k_r$ are the functions of M_value as Trend_Line in Fig. 11.

The weighted average method is used for defuzzification, and magnetometer error m_{fuzzy} is obtained as follows:

$$m_{\text{fuzzy}} = \frac{\sum_{i=1}^{r} \alpha_1 m_{ei}}{\sum_{i=1}^{r} \alpha_i} \quad (9)$$

where $i = 1, \ldots, r$. If θ_{set} meets a fuzzy rule, let the firing strength be $\alpha_i = 1$; otherwise, let $\alpha_i = 0$.

Finally, the magnetometer value $\theta_{\mathrm{Mf}}(n)$ is the sum of absolute direction θ_M and m_{fuzzy} of the magnetometer, i.e.,

$$\theta_{\mathrm{Mf}}(n) = \theta_M(n) + m_{\mathrm{fuzzy}}. \tag{10}$$

For corrected angles $\theta_{\mathrm{Gf}}(n)$ and $\theta_{\mathrm{Mf}}(n)$ of the gyroscope and magnetometer, the regular errors of the sensors are corrected using the method of "fuzzy compensation." The remaining irregular errors are not caused by the sensors but by the Trend_Line, which are irregular errors. The irregular errors are estimated by the Kalman filter algorithm. As the "fuzzy compensation" method is applied on both sensors in the same manner, the ratio of $\theta_{\mathrm{Gf}}(n)$ and $\theta_{\mathrm{Mf}}(n)$ is taken as 5:5. The direction angle of the robot can be obtained, as follows:

$$\theta_{\mathrm{now}}(n) = \theta_f(n) = \frac{1}{2}\left[\theta_{\mathrm{Gf}}(n) + \theta_{\mathrm{Mf}}(n)\right]. \tag{11}$$

3) Kalman Filter: To weaken the electromagnetic interference and the other irregular noise due to unavoidable reasons, after calculating g_{fuzzy} and m_{fuzzy} by the fuzzy algorithm, these two values and a Kalman filter are used to remove irregular errors. The Kalman filter is a recursive estimation algorithm, which knows the estimated value of the previous state and the observation value of the current state. Then, calculate the minimum mean square error as the gain of the Kalman k_gain. The optimum angle can be predicted, as well as the deviation corrected according to the measured values of the previous time corresponding to the covariance p_1. The measured value of each iteration corresponds to the covariance p to obtain the estimation values of the current state. The greatest difference between the Kalman filter and other algorithms is that the former is a time-domain filter; it does not need to transfer the time domain to the frequency domain. The comparison table of Kalman parameters and experiment symbols is shown in Table I.

The Kalman filter model is as follows:

$$\text{system state equation}: x_{k|k} = A_k x_{k-1|k-1} + B_k W_k \tag{12}$$

$$\text{measurement equation}: M_{k|k} = C_k x_{k|k-1} + V_k \tag{13}$$

$$\text{Predict}: \begin{cases} \hat{x}_{(k|k-1)} = A_k \hat{x}_{(k-1|k-1)} \\ p_{(k|k-1)} = A_k p_{(k-1|k-1)} A_k^T + Q_k \end{cases} \tag{14}$$

$$\text{Update}: \begin{cases} \hat{x}_{(k|k)} = \hat{x}_{(k|k-1)} + K_g\left(M_{(k|k)} - C_k \hat{x}_{(k|k-1)}\right) \\ K_g = \frac{\left(p_{k|k-1} C_k^T\right)}{\left(C_k p_{k|k-1} C_k^T + R_k\right)} \\ p_{(k|k)} = (1 - K_g C_k) p_{(k|k-1)}. \end{cases} \tag{15}$$

When the system goes to the next state, let $k = k + 1$, where $p_{k|k}$ is the next state of the $p_{k-1|k-1}$, and the algorithm recursive.

B. Specified Position Tracking Algorithm

$p_{\mathrm{now}}(x_{\mathrm{now}}, y_{\mathrm{now}})$ is the position coordinate of the robot when it is in the nth state and can be divided into four regions. Angle coordinates are represented by the coordinates of the magnetometer shown in Fig. 8. In order to verify the feasibility of the localization, the p_{now} and $p_{\mathrm{sp}}(x_b, y_b)$ are used to calculate the $\theta_{\mathrm{sp}}(n), l_b, \theta_{\mathrm{rotate}}$ and the direction of the robot's rotation as the next action commands for the robot. n is the nth period for the update angle.

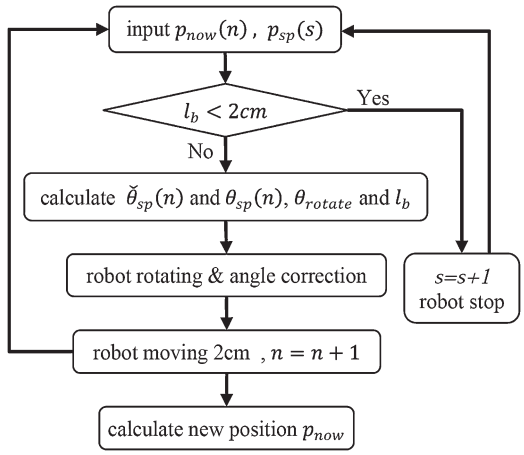

Fig. 6. Specified position tracking.

When $0° \le \theta_{\mathrm{now}}(n) < 90°$

$$x_{\mathrm{now}}(n) = x_{\mathrm{now}}(n-1) + l_b \cdot \cos\left((\theta_{\mathrm{now}}(n)\right)$$
$$y_{\mathrm{now}}(n) = y_{\mathrm{now}}(n-1) - l_b \cdot \sin\left((\theta_{\mathrm{now}}(n)\right).$$

When $90° \le \theta_{\mathrm{now}}(n) < 180°$

$$x_{\mathrm{now}}(n) = x_{\mathrm{now}}(n-1) - l_b \cdot \cos\left((180° - \theta_{\mathrm{now}}(n)\right)$$
$$y_{\mathrm{now}}(n) = y_{\mathrm{now}}(n-1) - l_b \cdot \sin\left((180° - \theta_{\mathrm{now}}(n)\right).$$

When $180° \le \theta_{\mathrm{now}}(n) < 270°$

$$x_{\mathrm{now}}(n) = x_{\mathrm{now}}(n-1) - l_b \cdot \cos\left((180° + \theta_{\mathrm{now}}(n)\right)$$
$$y_{\mathrm{now}}(n) = y_{\mathrm{now}}(n-1) + l_b \cdot \sin\left((180° + \theta_{\mathrm{now}}(n)\right).$$

When $270° \le \theta_{\mathrm{now}}(n) < 360°$

$$x_{\mathrm{now}}(n) = x_{\mathrm{now}}(n-1) + l_b \times \cos\left((360° - \theta_{\mathrm{now}}(n)\right)$$
$$y_{\mathrm{now}}(n) = y_{\mathrm{now}}(n-1) + l_b \times \sin\left((360° - \theta_{\mathrm{now}}(n)\right). \tag{16}$$

1) Angle Correction Period: The angle correction period is set for the robot, which moves 2 cm at each action until the distance between the robot and the specified position is less than 2 cm, and when $l_b < 2$ cm, it represents that the robot has reached the specified position. This method can ensure that, when the robot arrives at the position, the entire trajectory did not easily deviate. The flowchart is shown in Fig. 6, where s is the sth period for update position.

2) Relative Direction Angle: The current position p_{now} and the specified position p_{sp} are calculated by inverse trigonometric functions (arctan), which gives us the relative direction angles of specified position $\breve{\theta}_{\mathrm{sp}}(n) = \tan^{-1}|(y_b - y_{\mathrm{now}})|/|(x_b - x_{\mathrm{now}})|$, $\breve{\theta}_{\mathrm{sp}}(n) \in [0°, 90°]$ (see Fig. 7). In order to convert $\breve{\theta}_{\mathrm{sp}}(n)$ to an absolute angle, the relationship of the relative position between p_{now} and p_{sp} is analyzed as follows. The front

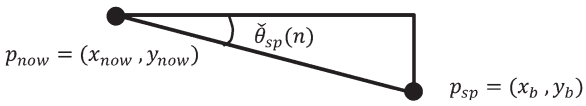

Fig. 7. Definition of $\breve{\theta}_{\mathrm{sp}}(n)$.

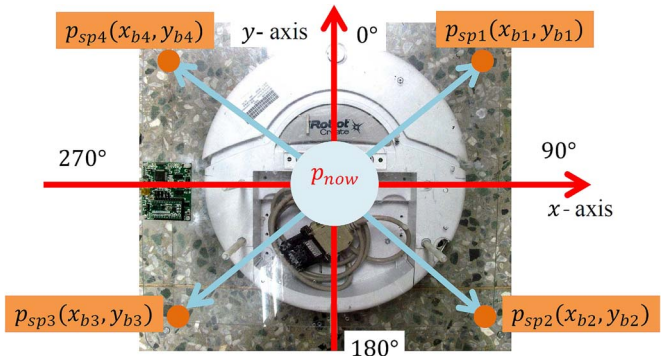

Fig. 8. Distribution diagram of p_{sp} and p_{now}.

TABLE II
CALCULATIONS OF $\theta_{\mathrm{sp}}(n)$

Location of p_{sp} & p_{now}	Formula of absolute angle
$x_{b1} > x_{now}, y_{b1} > y_{now}$	$\theta_{sp}(n) = 90° - \breve{\theta}_{sp}(n)$
$x_{b2} > x_{now}, y_{b2} < y_{now}$	$\theta_{sp}(n) = 90° + \breve{\theta}_{sp}(n)$
$x_{b3} < x_{now}, y_{b3} < y_{now}$	$\theta_{sp}(n) = 270° - \breve{\theta}_{sp}(n)$
$x_{b4} < x_{now}, y_{b4} > y_{now}$	$\theta_{sp}(n) = 270° + \breve{\theta}_{sp}(n)$

orientation of the robot is set to 0°, and p_{sp} are located at four intervals: [0°, 90°], [90°, 180°], [270°, 360°], or [180°, 270°], as shown in Fig. 8. The absolute direction angles $\theta_{sp}(n)$ are shown in Table II.

The rotation angle of the robot θ_{rotate} (17) can be obtained by taking the difference between $\theta_{\mathrm{sp}}(n)$ and $\theta_{\mathrm{now}}(n)$ and by determining the direction of rotation according to the size of the difference and using the difference to change the angle and direction of rotation of the robot. Consider the localization error when the rotation angle of the robot is less than $2°(\theta_{\mathrm{rotate}} < 2°)$; it is deemed to have been aligned with the absolute direction angle of the specified position and the robot's straight action, i.e.,

$$\theta_{\mathrm{rotate}} = \begin{cases} \theta_{\mathrm{sp}} - \theta_{\mathrm{now}}, & \text{turn right, } 2° < \theta_{\mathrm{sp}} - \theta_{\mathrm{now}} \le 180° \\ 360° - (\theta_{\mathrm{sp}} - \theta_{\mathrm{now}}), & \text{turn left, } 180° < \theta_{\mathrm{sp}} - \theta_{\mathrm{now}} \le 358° \\ \theta_{\mathrm{sp}} - \theta_{\mathrm{now}}, & \text{turn left, } 2° < \theta_{\mathrm{now}} - \theta_{\mathrm{sp}} \le 180° \\ 360° - (\theta_{\mathrm{now}} - \theta_{\mathrm{sp}}), & \text{turn right, } 180° < \theta_{\mathrm{now}} - \theta_{\mathrm{sp}} \le 358° \\ 0°, & \text{run straight, otherwise.} \end{cases} \tag{17}$$

l_b, θ_{rotate}, and the direction of rotation are movement commands for the next action of the robot, i.e.,

$$l_b = \sqrt{(x_b - x_{\mathrm{now}})^2 + (y_b - x_{\mathrm{now}})^2}. \tag{18}$$

Let $g_{\mathrm{angle}} = \theta_{\mathrm{rotate}}$, and the action recursive is $(n = n + 1)$. The new absolute direction angle θ_{now} can be calculated by angle correction algorithms (see Fig. 4), then it was substituted into (16) to obtain the new position p_{now}, and then execute the next position tracking.

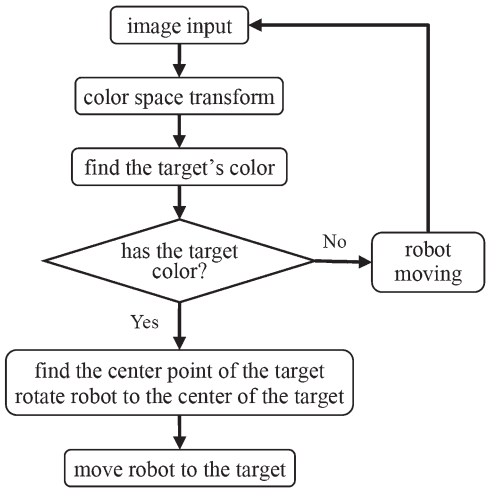

Fig. 9. Flowchart of the image processing method.

TABLE III
SETTING AND ACTUAL MOVING DISTANCE (UNIT: CENTIMETERS)

set distance	2	5	10	50
average distance	1.63	4.6	9.75	49.7
error average	0.37	0.4	0.25	0.3

C. Image Recognition and Object Avoidance

Fig. 9 describes the block diagram of the proposed image recognition and tracking system. YCbCr color space is used to detect the target object colors. A Kinect camera is used to capture the object, which is selected by the user in our human–machine interface. When the designated object is detected by the image processing function, the robot is controlled and moves in on the object, according to the image tracking information.

IV. EXPERIMENTAL AND RESULT

A. Analysis of Error Angle and Distance

In order to obtain the error information of the angle and distance to be traversed, two experiments are designed regarding the error analysis of the traversed distance and rotation angle.

1) Moving Distance Error Analysis: After making the robot move forward and record moving distance sequentially, compare the differences between the actual moving distance and the set distances. The forward setting distances were 2, 5, 10, and 50 cm, and each set distance test was run ten times for a total of 40 data. The smallest accurate scale of an experimental measurement tool is 0.1 cm. The results are shown in Table III. Experimental results show that the gap between the set distance and the real distance is tiny. Take the values of the speedometer directly from the one built into the iRobot and use them to calculate the distance l_b.

2) Rotation Angle Error Analysis: An experiment regarding the rotation angle of the robot was designed, and the sensor values from the gyroscope and magnetometer were analyzed

TABLE IV
EXPERIMENTAL DATA OF ROBOT TURNING RIGHT

set_angle	10	20	30	40	50	60	70	80	90
actual _value	9.75	20.15	29.95	39.75	50.05	60.45	70.70	81.05	90.45
G_value	9.53	19.50	29.62	93.50	49.74	59.68	69.45	79.45	89.65
G error	0.77	0.95	1.25	0.99	1.06	1.10	1.65	1.85	1.33
Trend_Line	1.02	1.02	1.02	1.02	1.02	1.02	1.59	1.59	1.59
set-angle	100	110	120	130	140	150	160	170	180
actual _value	99.65	110.25	121	130.65	141.05	151.5	160.3	170.1	180.55
G_value	99.31	109.63	119.49	129.68	139.64	149.45	159.68	169.51	180.02
G_error	1.58	1.49	1.74	1.55	1.68	2.24	1.00	1.47	1.45
Trend_Line	1.59	1.59	1.59	1.59	1.59	1.59	1.59	1.59	1.59

TABLE V
EXPERIMENTAL DATA OF ROBOT TURNING LEFT

set-angle	190	200	210	220	230	240	250	260	270
actual _value	189.35	200	210.5	220.1	228.9	239.75	249.75	260.6	269.75
G_value	190.34	200.38	210.49	220.56	230.24	240.28	250.40	260.63	270.48
G error	-1.65	-1.27	-1.24	-1.26	-1.94	-1.40	-1.16	-1.41	-1.54
Trend_Line	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47
set_angle	280	290	300	310	320	330	340	350	-
actual _value	279.75	289	299.4	309.4	319.4	328.95	338	348.15	-
G_value	280.47	290.47	300.37	310.55	320.51	330.26	340.43	350.51	-
G error	-1.33	-1.86	-1.51	-1.38	-1.43	-1.65	-2.63	-2.67	-
Trend_Line	-1.47	-1.47	-1.47	-1.47	-1.47	-1.47	-2.65	-2.65	-

and recorded. The measuring range is 10°–350°. The robot angle to the right is defined as 10°–180°, whereas the angle to the left is defined as 190°–360°. The difference between the actual moving angle (actual_value) and the gyroscope value (G_value) is measured while the robot is rotating. After the robot stops rotating, the difference between the actual moving angle (actual_value) and the magnetometer value (M_value) is measured. During the experiment, the moving speed is fixed at 150 mm/s, and the laptop will finish calculating the angle values of the gyroscope and magnetometer when the robot stops rotating. The given values, the set angles, and the actual rotation values are compared. The precise scale of the measuring tool is 1°.

a) Gyroscope error analysis: From 10° to 350°, The interval of set_angle is 10° and the experiment is done ten times per interval, each time recording a gyroscope value, for a total of 350 pieces of data. The values for each experimental data, calculated by averaging, are shown in Tables IV and V (unit: degree). "G_error" refers to the error between the actual rotation angle and the gyroscope value. Fig. 10 shows the G_error curve and its Trend_Line. Formulating Trend_Line to fuzzy rules to speculate the error of gyroscope and then use the result to eliminate the error of gyroscope g_fuzzy.

b) Magnetometer error analysis: From 10° to 350°, the interval of set_angle is 10°, and the experiment is done ten times per interval, each time recording 21 magnetometer values, for a total of 7350 pieces of data. The values of each experimental data, calculated by averaging, are shown in Tables VI and VII (unit: degree). "M_error" refers to the error between the actual rotation angle and the magnetometer value. Fig. 11 shows the M_error curve and its Trend_Line. Formulate the Trend_Line to fuzzy rules to speculate the error of gyroscope and then use the result to eliminate the error of magnetometer m_fuzzy.

B. Algorithms to Achieve

1) Statistical Weighting: The value of the Trend_Line can be calculated for an approximate value of the integral error of the gyroscope or magnetometer. It is estimated from the error value (G_error or M_error), and the error value is the difference between the actual robot rotation angle and the gyroscope (or magnetometer) angle. The error integral is to calculate the error area in Figs. 10 and 11. The computing mode is to add the products of all the Trend_Line values and angle (10°). The formula of the weighting value is given by

$$W_{\text{value}} = |T_{\text{value}}| * N * 10^{\circ} \tag{19}$$

where T_{value} is the value of the Trend_Line, and N is the number of Trend_Line when the Trend_Line's value is T_{value}. The error integral of the gyroscope is $525.5 \approx 526$, as obtained by multiplying all Trend_Line values in Tables IV and V by (each set angular interval). The error integral of magnetometer is 4840, as obtained by multiplying all Trend_Line values in Tables VI and VII by 10° (each set angular interval). The weighted ratio is inversely proportional to the error integral, i.e., the weighting of the gyroscope. The weighting of the magnetometer $= 4840$: $526 \approx 9 : 1$. The current rotation angle is given as

$$\theta_{\text{now}}(n) = \theta_w(n) = \theta_G(n) \times 0.9 + \theta_M(n) \times 0.1. \tag{20}$$

2) Fuzzy Compensation: In Figs. 10 and 11, the fuzzy rules of the gyroscope and magnetometer were determined, respectively, as follows.

a) Gyroscope error fuzzy rules: Fig. 10 shows the error curve of the gyroscope and the actual rotation angle of the robot. In order to obtain the fuzzy rules, the Trend_line is used to approximate the error curve. It is observed in Fig. 10 that the Trend_line has five bend line segments, and the fuzzy rule base ($r1$ to $r5$) is defined by the five bend line segments. The "premise" of the fuzzy rule base defines which interval of the G_value the measured angle g_{angle} is located in, and the "conclusion" is the Trend_line value in the interval. The gyroscope fuzzy rule base of this experiment is described, as follows:

$$\begin{aligned}
&r1: \text{If } 0^{\circ} \le g_{\text{angle}} \le 80^{\circ}, \quad \text{Then } g_{e1} = 1.0202\\
&r2: \text{If } 50^{\circ} \le g_{\text{angle}} \le 180^{\circ}, \quad \text{Then } g_{e2} = 1.5859\\
&r3: \text{If } 180^{\circ} \le g_{\text{angle}} \le 200^{\circ}, \quad \text{Then } g_{e3} = -0.2959 * G_\text{value} + 54.8539\\
&r4: \text{If } 190^{\circ} \le g_{\text{angle}} \le 340^{\circ}, \quad \text{Then } g_{e4} = -1.468\\
&r5: \text{If } 320^{\circ} \le g_{\text{angle}} \le 360^{\circ}, \quad \text{Then } g_{e5} = -2.65.
\end{aligned} \tag{21}$$

The weighted average method is used for defuzzification. If g_{angle} meets a fuzzy rule, let the firing strength $\alpha_1 = 1$; otherwise, let $\alpha_1 = 0$. The following is the defuzzification

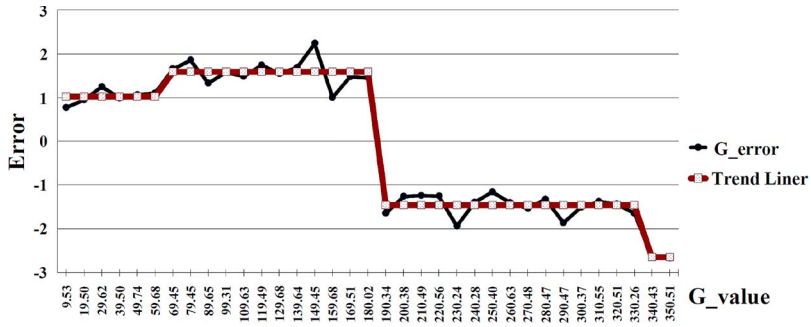

Fig. 10. Gyroscope and actual rotation-angle error.

TABLE VI
EXPERIMENTAL DATA OF ROBOT TURNING RIGHT

set_angle	10	20	30	40	50	60	70	80	90
actual _value	9.75	20.15	29.95	39.75	50.05	60.45	70.70	81.05	90.45
M_value	2.26	9.77	20.72	33.16	46.99	60.99	75.70	90.31	105.17
M_error	8.01	10.92	9.90	7.62	4.52	-3.14	-7.27	-9.55	-14.78
Trend_Line	9.9	9.9	9.9	6.38	2.46	-1.5	-5.66	-9.8	-14.01
set_angle	100	110	120	130	140	150	160	170	180
actual _value	99.65	110.25	121.00	130.65	141.05	151.50	160.30	170.10	180.10
M_value	118.42	131.79	144.57	156.02	167.37	177.72	187.72	197.42	207.19
M_error	-18.78	-21.54	-23.47	-25.40	-26.37	-26.26	-27.48	-27.38	-26.70
Trend_Line	-17.75	-21.54	-23.27	-24.83	-26.76	-26.76	-26.76	-26.76	-26.76

TABLE VII
EXPERIMENTAL DATA OF ROBOT TURNING LEFT

set_angle	190	200	210	220	230	240	250	260	270
actual _value	189.35	200	210.05	220.10	228.90	239.75	249.75	260.60	269.75
M_value	215.65	224.58	232.59	240.89	247.88	255.88	263.47	271.28	278.18
M_error	-26.36	-24.65	-22.60	-20.85	-19.06	-16.21	-13.79	-12.40	-8.57
Trend_Line	-26.76	-23.82	-21.54	-19.18	-17.19	-14.91	-12.75	-10.53	-8.57
set_angle	280	290	300	310	320	330	340	350	-
actual _value	279.75	289.00	299.40	309.40	319.40	328.95	338.00	348.15	-
M_value	285.34	292.34	299.67	307.07	314.54	321.92	330.23	338.89	-
M_error	-5.72	-3.49	-0.89	2.41	4.90	7.07	7.84	9.33	-
Trend_Line	-6.31	-4.1	-1.8	0.54	2.89	5.21	7.84	9.33	-

equation, where the gyroscope error g_{fuzzy}, as calculated by fuzzy compensation, can be obtained:

$$g_{\text{fuzzy}} = \frac{\sum_{i=1}^{n} \alpha_i g_{ei}}{\sum_{i=1}^{n} \alpha_i}. \tag{22}$$

The gyroscope value $\theta_{\text{Gf}}(n)$, as calculated by fuzzy compensation, is the sum of the absolute angle $\theta_G(n)$ and g_{fuzzy} of the gyroscope, i.e.,

$$\theta_{\text{Gf}}(n) = \theta_G(n) + g_{\text{fuzzy}}. \tag{23}$$

b) Magnetometer error fuzzy rules: Fig. 11 shows the error curve of the magnetometer and the actual rotation angle of the robot. In order to obtain fuzzy rules, the Trend_Line is used to approximate the error curve. It is observed in Fig. 11 that there are seven bend line segments and 338.89° to 360°; thus, there are eight intervals. The fuzzy rule base ($r1$ to $r8$) is defined by the eight bend line segments. The premise of fuzzy rules defines which interval of the M_value the measured angle of θ_{set} is located in, and the conclusion is the Trend_Line value in the interval. The magnetometer fuzzy rule base of this experiment is described as follows:

$$\begin{aligned}
&r1: \text{ If } 0^\circ \le \theta_{\text{set}} \le 20.72^\circ \quad \text{Then } m_{e1} = 9.898\\
&r2: \text{ If } 20.72^\circ \le \theta_{\text{set}} \le 131.79^\circ\\
&\qquad \text{Then } m_{e2} = -0.283 * M_\text{value} + 15.76\\
&r3: \text{ If } 131.79^\circ \le \theta_{\text{set}} \le 167.37^\circ\\
&\qquad \text{Then } m_{e3} = -0.136 * M_\text{value} - 3.65\\
&r4: \text{ If } 167.37^\circ \le \theta_{\text{set}} \le 215.65^\circ \quad \text{Then } m_{e4} = -26.76\\
&r5: \text{ If } 215.65^\circ \le \theta_{\text{set}} \le 278.18^\circ\\
&\qquad \text{Then } m_{e5} = 0.284 * M_\text{value} - 87.74\\
&r6: \text{ If } 278.18^\circ \le \theta_{\text{set}} \le 330.23^\circ\\
&\qquad \text{Then } m_{e6} = 0.315 * M_\text{value} - 96.23\\
&r7: \text{ If } 330.23^\circ \le \theta_{\text{set}} \le 338.89^\circ\\
&\qquad \text{Then } m_{e7} = 0.173 * M_\text{value} - 49.30\\
&r8: \text{ If } 338.89^\circ \le \theta_{\text{set}} \le 360^\circ \quad \text{Then } m_{e8} = 9.33.
\end{aligned} \tag{24}$$

The weighted average method is used for defuzzification. If θ_{set} meets a fuzzy rule, let the firing strength $\alpha_i = 1$; otherwise, let $\alpha_i = 0$. The following is the defuzzification equation, where magnetometer error m_{fuzzy}, as calculated by fuzzy compensation, can be obtained:

$$m_{\text{fuzzy}} = \frac{\sum_{i=1}^{n} \alpha_i m_{ei}}{\sum_{i=1}^{n} \alpha_i}. \tag{25}$$

The magnetometer value $\theta_{\text{Mf}}(n)$, as calculated by fuzzy compensation, is the sum of absolute direction angle $\theta_M(n)$ and m_{fuzzy} of the magnetometer, i.e.,

$$\theta_{\text{Mf}} = \theta_M(n) + m_{\text{fuzzy}}. \tag{26}$$

Finally,

$$\theta_{\text{now}}(n) = \theta_f(n) = \frac{1}{2}\left[\theta_{\text{Gf}}(n) + \theta_{\text{Mf}}(n)\right]. \tag{27}$$

3) Fuzzy + Kalman Filter: The g_{fuzzy} and m_{fuzzy}, as obtained by fuzzy compensation, are imported into the Kalman filter algorithm. The parameters of the Kalman filter are shown in Table VIII, where the number of iterations $j = 41$, $i = \{1, 2, \ldots, j\}$ represents the i iteration. A constant $c \in [0, 1]$ for which the size is associated with the speed of Kalman

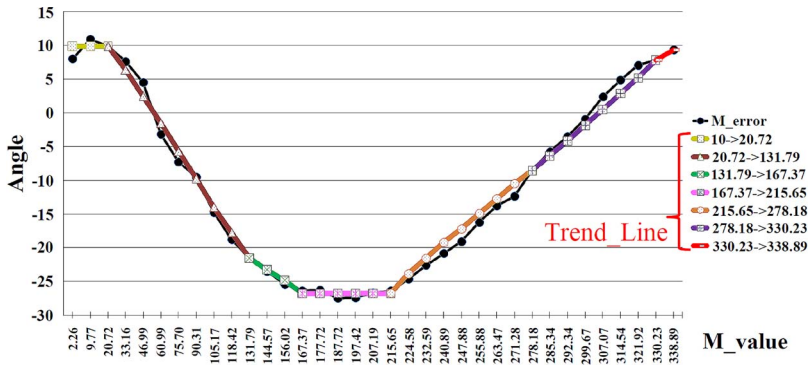

Fig. 11. Magnetometer and actual rotation-angle error.

TABLE VIII
KALMAN FILTER PARAMETERS

Kalman Filter	Definition
$g_{fuzzy}(i)$	Gyro error calculated by No. i fuzzy compensation
$m_{fuzzy}(i)$	Magnetometer error calculated by No. i fuzzy compensation
$\theta_{KG}(i)$	Gyroscope correction value of No. i iteration
$\theta_{kal}(i)$	Kalman correction value of No. i iteration (integrated with gyroscope and magnetometer)
$\theta_{kal}(40)$	The value of the final iteration of Kalman is taken as the optimal correction value
$p_1(i)$	Gyroscope predicted covariance of No. i iteration
$p(i-1)$	Gyroscope updated covariance of No. $i-1$ iteration
$p(i)$	Gyroscope updated covariance of No. i iteration
$\theta_{KM}(i)$	Magnetometer correction value of No. i iteration
k_gain	Optimal Kalman gain
g_{e_std}	Standard deviation of gyroscope correction values of 40 iterations
m_{e_std}	Standard deviation of magnetometer correction values of 40 iterations
$c = 0.5$	Optimal convergence constant
$j = 40$	Optimal number of iterations

convergence when $c = 0.5$; the optimal convergence rate can be obtained. Fig. 12 shows the Kalman filter algorithm process, which corresponds to the pseudocode in Table IX, where Random(-3.3) and Random$(-4.5, 4.5)$ are the optimal noise estimation range as obtained by the experiments. The standard deviation of g_{e_std} and m_{e_std} of $m_{\mathrm{fuzzy}}(i)$ and $g_{\mathrm{fuzzy}}(i)$ are used to calculate the range of irregular noise, i.e.,

$$g_{e\_\mathrm{std}} = \sqrt{\frac{\sum_{i=1}^{j}\left(g_{\mathrm{fuzzy}}(i) - \overline{g_{\mathrm{fuzzy}}}\right)^2}{j-1}} \tag{28}$$

$$m_{e\_\mathrm{std}} = \sqrt{\frac{\sum_{i=1}^{j}\left(m_{\mathrm{fuzzy}}(i) - \overline{m_{\mathrm{fuzzy}}}\right)^2}{j-1}} \tag{29}$$

where the $\overline{g_{\mathrm{fuzzy}}}$ and $\overline{m_{\mathrm{fuzzy}}}$ are the average values of the g_{fuzzy} and m_{fuzzy}, respectively.

In this paper, the Kalman filter is combined with fuzzy compensation in order to fuse the gyroscope and magnetometer error values. Fuzzy compensation is used to estimate the regular errors g_{fuzzy} and m_{fuzzy} of the gyroscope and magnetometer. The fuzzy rule base is determined by using the Trend_Line of

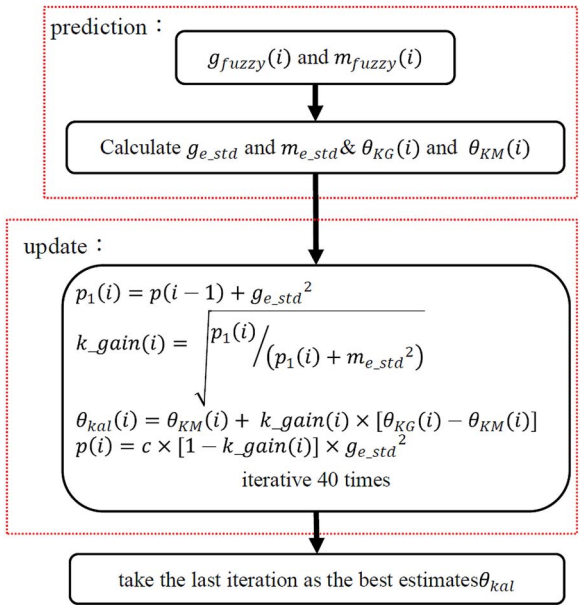

Fig. 12. Flowchart of the Kalman filter.

TABLE IX
EXPERIMENTAL DATA (STATISTICAL WEIGHTING)

Statistical Weighting						
Item	p_{sp}	θ_{sp}	p_{real}	θ_{real}	p_{now}	θ_{now}
dot 0	(0,39.5)	201.075°	(-2.9, 26.5)	188.505°	(0.7, 39.14)	202.39°
dot 1	(40,59.5)	333.435°	(37.1, 60.6)	330.372°	(38.21, 58.66)	333.47°
dot 2	(70,59.5)	358.486°	(69.2, 61)	359.286°	(69.49, 59.77)	352.05°
dot 3	(80,24.5)	73.407°	(78.6, 28.5)	73.869°	(79.84,25.29)	74.38°
dot 4	(50,19.5)	169.019°	(50.6, 18.5)	160.346°	(51.12, 19.8)	167.04°

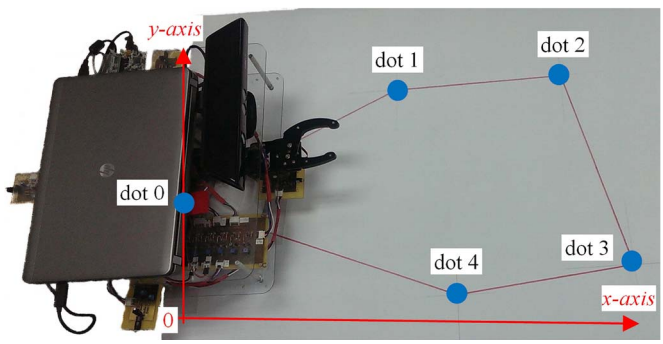

Fig. 13. Experimental environment.

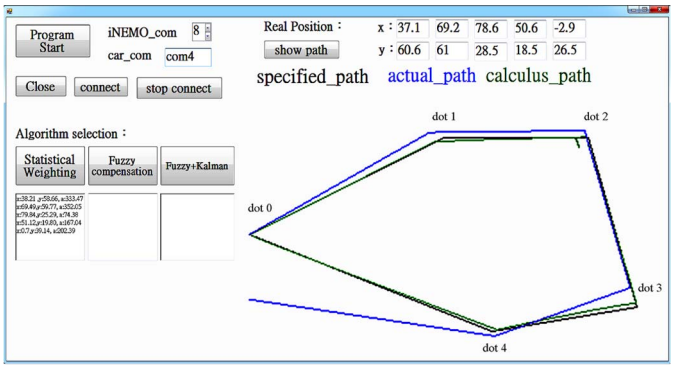

Fig. 14. iRobot moving path (statistical weighting).

TABLE X
EXPERIMENTAL DATA (FUZZY COMPENSATION)

Fuzzy compensation						
Item	p_{sp}	θ_{sp}	p_{real}	θ_{real}	p_{now}	θ_{now}
dot 0	(0,39.5)	201.143°	(-2,42.8)	203.321°	(0.76, 39.19)	201.91°
dot 1	(40,59.5)	333.435°	(41, 60.3)	333.101°	(38.2, 58.65)	335.2°
dot 2	(70,59.5)	358.469°	(72.1, 60)	0.553°	(69.68, 59.53)	359.91°
dot 3	(80,24.5)	73.585°	(78.4, 25.4)	79.681°	(79.88, 25.02)	74.2°
dot 4	(50,19.5)	169.533°	(48.8, 20.9)	171.356°	(51.12, 19.73)	168.47°

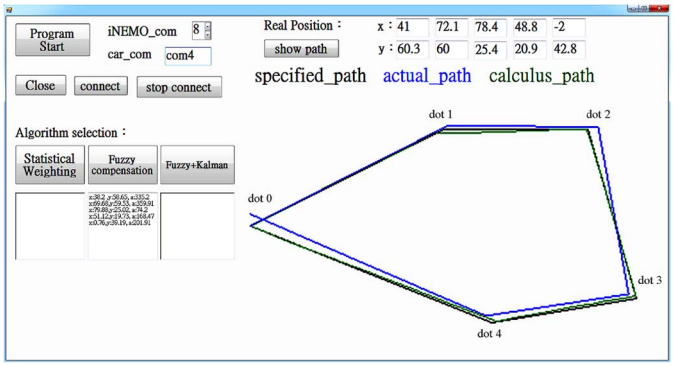

Fig. 15. iRobot moving path (fuzzy compensation).

TABLE XI
EXPERIMENTAL DATA (FUZZY + KALMAN)

Fuzzy + Kalman						
Item	p_{sp}	θ_{sp}	p_{real}	θ_{real}	p_{now}	θ_{now}
dot 0	(0,39.5)	201.115°	(-2.6, 42.2)	202.946°	(0.82, 39.22)	204.56°
dot 1	(40,59.5)	333.435°	(39, 59.8)	332.502°	(38.19, 58.69)	333.55°
dot 2	(70,59.5)	358.541°	(71, 60.3)	359.105°	(69.67, 59.52)	0.69°
dot 3	(80,24.5)	73.565°	(77, 25.5)	80.218°	(79.93, 25.04)	70.95°
dot 4	(50,19.5)	169.513°	(52.2, 19)	165.313°	(51.17, 19.74)	168.46°

Figs. 10 and 11. The Kalman filter adds random values in each iteration, e.g., Random(-3.3) in g_{fuzzy} of the gyroscope and Random$(-4.5, 4.5)$ in m_{fuzzy} of the magnetometer, whereas the irregular error is added in the estimated values. Finally, value $\theta_{\text{kal}}(40)$ of the 40th iteration of the Kalman filter is taken as the optimal correction value. Let $\theta_{\text{now}}(n) = \theta_{\text{kal}}(40)$.

C. Experimental Results

To verify whether the Kalman design and the fuzzy database rule are effective in reducing errors, this paper designed the

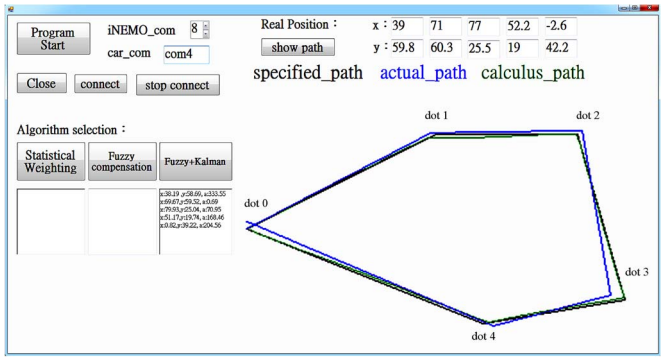

Fig. 16. iRobot moving path (fuzzy + Kalman).

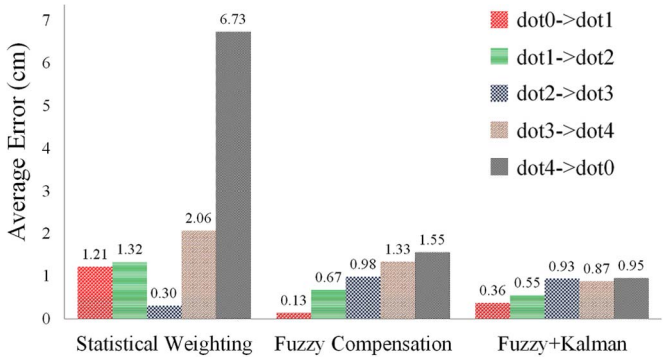

Fig. 17. Specified & actual path.

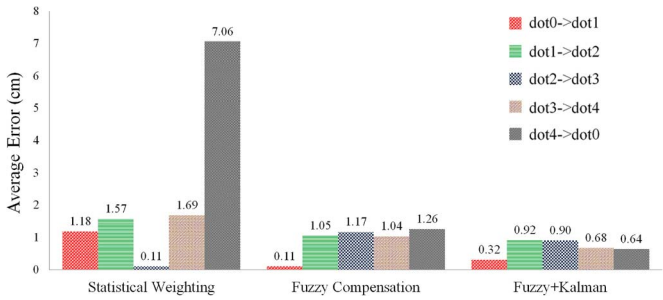

Fig. 18. Calculus & actual path.

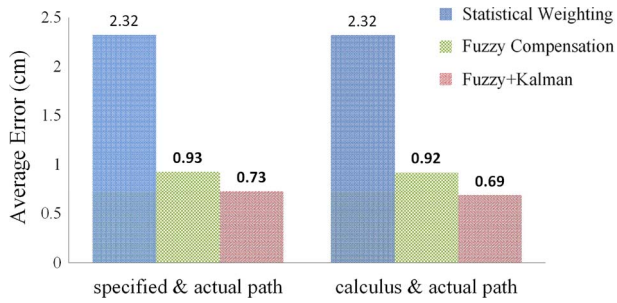

Fig. 19. Path error comparisons.

implementation, respectively. The actual range size of the map in our experiment is 109.1 cm × 78.6 cm. Finally, we compare the correction results of various algorithms that this paper proposed.

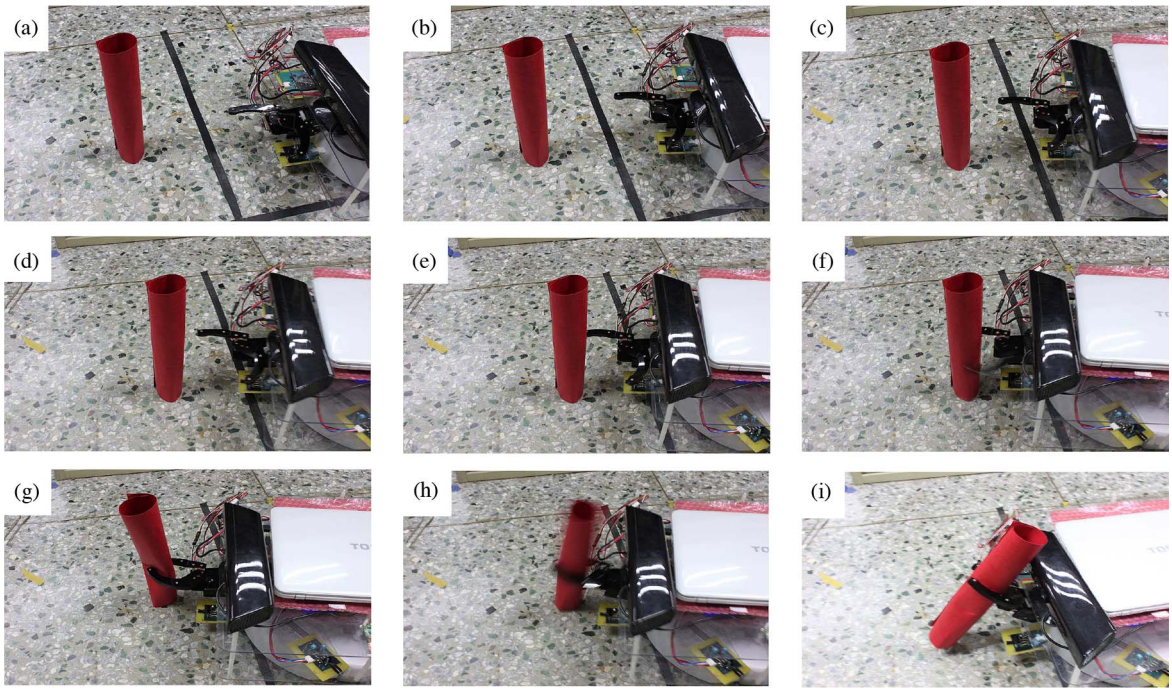

Fig. 20. Human–computer interface.

1) Localization Experiment: To make the robot move and rotate during the actual experiment, the rotation angle of the robot is measured and corrects the angle at the next rotation. The experimental environment is shown in Fig. 13. The coordinates of the five specific positions: dot 0 (0, 39.5), dot 1 (40, 59.5), dot 2 (70, 59.5), dot 3 (80, 24.5), and dot 4 (50, 19.5). After all the paths (dot 0 $\rightarrow$ dot 1 $\rightarrow$ dot 2 $\rightarrow$ dot 3 $\rightarrow$ dot 4) were finished, the actual position coordinates of the robot can also be measured manually; they need to be input into the program to draw the walking paths. Then, compare the errors of the specified position and angle (p_{sp} and θ_{sp}), to the actual position and angle (p_{real} and θ_{real}), and calculate the position and angle (p_{now} and θ_{now}).

a) Statistical weighting: The experimental data and moving path are shown in Table IX and Fig. 14; the effect of correction angle is not accurate, maybe even leading to location failure.

b) Fuzzy compensation: The experimental data and moving path are shown in Table X and Fig. 15; the effect of correction angle is better than a statistical weighting method, but the actual path still is offset from the specified path.

c) Fuzzy + Kalman filter: The experimental data and moving path are shown in Table XI and Fig. 16; the effect of correction angle is better than the statistical weighting method and better than fuzzy compensation in the path dot4 $\rightarrow$ dot0.

2) Path Comparison: This paper proposes three kinds of angle correction methods in order to compare the average error values of "specified & actual path" and "calculated & actual path," shown in Figs. 17 and 18. The method for getting these errors values lies in the vertical distance between the line of the specified path and the point on the actual path or the line of the calculus path and the point on the actual path. Fig. 19 shows the comparison of average path error values of the three kinds of algorithms. The algorithm "Fuzzy + Kalman" comprise a more accurate correction algorithm compared with the others.

D. Image Recognition, Collision Avoidance, and Object Catching

Fig. 20 shows the forward motion of the robot. The IR sensor at the front of the robot can keep sensing the distance between the robot and target object. When the robot (IR sensor) closes in on the target object at a short distance (5 cm), the distance change information will transmit to the laptop via the DSP chip. Then, the laptop will send the command to the DSP chip to generate two PWM signals to control the mechanical arm. The first PWM signal is used to clamp (or release) the object, whereas the second PWM signal is used to lift up the object.

V. Conclusion

This paper has developed an indoor intelligent service mobile robot that can locate position, avoid obstacles, use image recognition, and grip the target object. The main focus of this paper is to establish a precise relative position localization system. After the user sets the robot's initial and specified positions, the robot's position can be calculated by the distance and direction angle after the completion of each robot action. After the fuzzy compensation algorithm, the Kalman filter was applied and then the former was used to eliminate the regular error and the latter was used to eliminate the irregular error. In order to verify the feasibility of localization, setting the angle of the correction period for the robot to move a short distance (2 cm) until the robot is very close to the specified position (2 cm) represents reaching the specified position. Experimental results show that fuzzy compensation combined with a Kalman filter algorithm is effective.

REFERENCES

[1] [Online]. Available: http://www.irobot.com/

[2] [Online]. Available: http://www.funrobot.com/zh-tw/about.html

[3] G. M. Song, K. J. Yin, Y. X. Zhou, and X. Z. Cheng, "A surveillance robot with hopping capabilities for home security," *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 2034–2039, Nov. 2009.

[4] C. C. Tseng, C. L. Lin, B. Y. Shih, and C. Y. Chen, "SIP-enabled surveillance patrol robot," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 2, pp. 394–399, Apr. 2013.

[5] B. Graf, U. Reiser, M. Haägele, K. Mauz, and P. Klein, "Robotic home assistant Care-O-bot 3—product vision and innovation platform," in *Proc. IEEE Workshop ARSO*, Tokyo, Japan, Nov. 23–25, 2009, pp. 139–144.

[6] [Online]. Available: http://www.care-o-bot.de/english/index.php

[7] [Online]. Available: http://www.northropgrumman.com/Capabilities/Remotec/Pages/default.aspx

[8] J. Y. Gao *et al.*, "Heavy explosive removing robot control technique research," in *Proc. IEEE Int. Conf. IHMSC*, Hangzhou, China, Aug. 26–27, 2009, pp. 85–89.

[9] R. R. Murphy, J. Kravitz, S. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 91–103, Jun. 2009.

[10] Y. W. Li, S. R. Ge, H. Zhu, H. F. Fang, and J. K. Gao, "Mobile platform of rocker-type coal mine rescue robot," *Mining Sci. Technol. (China)*, vol. 20, no. 3, pp. 466–471, May 2010.

[11] S. X. Yang, A. M. Zhu, G. F. Yuan, and M. Q.-H. Meng, "A bioinspired neurodynamics-based approach to tracking control of mobile robots," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3211–3220, Aug. 2012.

[12] H. Xu and Y. P. Shen, "Target tracking control of mobile robot in diversified manoeuvre modes with a low cost embedded vision system," *Ind. Robot*, vol. 40, no. 3, pp. 275– 287, 2013.

[13] R. C. Luo and C. C. Lai, "Enriched indoor map construction based on multisensor fusion approach for intelligent service robot," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3135–3145, Aug. 2012.

[14] J.-S. Chiang, C.-H. Hsia, and H.-W. Hsu, "A stereo vision-based self-localization system," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1677–1689, May 2013.

[15] B. R. Sahraei, F. Shabaninia, A. Nemati, and S. D. Stan, "A novel robust decentralized adaptive fuzzy control for swarm formation of multiagent systems," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3124–3134, Aug. 2012.

[16] H. Hur and H.-S. Ahn, "Discrete-time filtering for mobile robot localization using wireless sensor network," *IEEE Sensors J.*, vol. 13, no. 1, pp. 245–252, Jan. 2013.

[17] C.-H. Ou and W.-L. He, "Path planning algorithm for mobile anchor-based localization in wireless sensor networks," *IEEE Sensors J.*, vol. 13, no. 2, pp. 466–475, Feb. 2013.

[18] B. Wang, S. Zhou, W. Liu, and Y. Mo, "Indoor localization based on curve fitting and location search using received signal strength," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 572–582, Jan. 2015.

[19] J. Wang *et al.*, "Lightweight robust device-free localization in wireless networks," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5681–5689, Oct. 2014.

[20] W. Wang and G. Xie, "Online high-precision probabilistic localization of robotic fish using visual and inertial cues," *IEEE Trans. Ind. Electron.*, vol. vol. 62, no. 2, pp. 1113–1124, Feb. 2015.

[21] B. Wu and C. Jen, "Particle-filter-based radio localization for mobile robots in the environments with low-density WLAN APs," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6860–6870, Dec. 2014.

[22] [Online]. Available: http://www.st.com/Internet/analog/family/89.jsp

[23] T. S. Li, S. J. Chang, and W. Tong, "Fuzzy target tracking control of autonomous mobile robots by using infrared sensors," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 491–501, Aug. 2004.

[24] C. F. Juang and Y. C. Chang, "Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 379–392, Apr. 2011.

[25] S. S. Han, D. K. Kim, and J. M. Lee, "A new tag arrangement pattern for a differential driving mobile robot based on RFID system," in *Proc. IEEE Int. Conf. ICCAS*, Seoul, Korea, Oct. 17–20, 2007, pp. 1228–1233.

[26] Y. Ma, L. Zhou, K. Liu, and J. Wang, "Iterative phase reconstruction and weighted localization algorithm for indoor RFID-based localization in NLOS environment," *IEEE Sensors J.*, vol. 14, no. 2, pp. 597–611, Feb. 2014.

[27] P. Yang and W. Wu, "Efficient particle filter localization algorithm in dense passive RFID tag environment," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5641–5651, Oct. 2014.

[28] Z. Zhang *et al.*, "Item-level indoor localization with passive UHF RFID based on tag interaction analysis," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2122–2135, Apr. 2014.

[29] E. DiGiampaolo and F. Martinelli, "Mobile robot localization using the phase of passive UHF RFID signals," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 365–376, Jan. 2014.

[30] Y. Wang, Y. Liu, H. Leung, R. Chen, and A. Li, "A multi-bit identification protocol for RFID tag reading," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3527–3536, Oct. 2013.

[31] G. Cicirelli, A. Milella, and D. D. Paola, "RFID tag localization by using adaptive neuro-fuzzy inference for mobile robot applications," *Ind. Robot*, vol. 39, no. 4, pp. 340–348, 2012.

[32] Q. X. Yu, C. Yuan, Z. Fu, and Y. Z. Zhao, "An autonomous restaurant service robot with high positioning accuracy," *Ind. Robot*, vol. 39, no. 3, pp. 271–281, 2012.

[33] R. Ogawara, M. Fujii, and Y. Watanabe, "A study on location tracking system using kalman filter based on sensor information," in *Proc. IEEE Int. Conf. ISITA*, Honolulu, HI, USA, Oct. 28–31, 2012, pp. 184–188.

[34] F. Rivard, J. Bisson, F. Michaud, and D. Létourneau, "Ultrasonic relative positioning for multi-robot systems," in *Proc. IEEE Int. Conf. ICRA*, Pasadena, CA, USA, May 19–23, 2008, pp. 323–328.

[35] A. Milano, A. Priolo, A. Gasparri, M. D. Rocco, and G. Ulivi, "An experimental validation of a low-cost indoor relative position localizing system for mobile robotic networks," in *Proc. 19th IEEE MED*, Corfu, Greece, Jun. 20–23, 2011, pp. 169–174.

[36] K. H. Choi, W.-S. Ra, S.-Y. Park, and J. B. Park, "Robust least squares approach to passive target localization using ultrasonic receiver array," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 1993–2002, Apr. 2014.

[37] O. De Silva, G. K. I. Mann, and R. G. Gosine, "An ultrasonic and vision-based relative positioning sensor for multirobot localization," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1716–1726, Mar. 2015.

[38] M. Narasimhappa, P. Rangababu, S. L. Sabat, and J. Nayak, "A modified sage-husa adaptive kalman filter for denoising fiber optic gyroscope signal," in *Proc. IEEE Int. Conf. INDICON*, Kochi, India, Dec. 7–9, 2012, pp. 1266–1271.

Authors' photographs and biographies not available at the time of publication.