

## Research Article

# Camera Space Particle Filter for the Robust and Precise Indoor Localization of a Wheelchair

Raul Chavez-Romero,<sup>1</sup> Antonio Cardenas,<sup>1</sup> Mauro Maya,<sup>1</sup>  
Alejandra Sanchez,<sup>1</sup> and Davide Piovesan<sup>2</sup>

<sup>1</sup>Universidad Autonoma de San Luis Potosi, 78290 San Luis Potosi, SLP, Mexico

<sup>2</sup>Gannon University, Erie, PA 16541, USA

Correspondence should be addressed to Raul Chavez-Romero; raulcr2010@gmail.com

Received 19 December 2014; Accepted 25 March 2015

Academic Editor: Changhai Ru

Copyright © 2016 Raul Chavez-Romero et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the theoretical development and experimental implementation of a sensing technique for the robust and precise localization of a robotic wheelchair. Estimates of the vehicle's position and orientation are obtained, based on camera observations of visual markers located at discrete positions within the environment. A novel implementation of a particle filter on camera sensor space (Camera-Space Particle Filter) is used to combine visual observations with sensed wheel rotations mapped onto a camera space through an observation function. The camera space particle filter fuses the odometry and vision sensors information within camera space, resulting in a precise update of the wheelchair's pose. Using this approach, an inexpensive implementation on an electric wheelchair is presented. Experimental results within three structured scenarios and comparative performance using an Extended Kalman Filter (EKF) and Camera-Space Particle Filter (CSPF) implementations are discussed. The CSPF was found to be more precise in the pose of the wheelchair than the EKF since the former does not require the assumption of a linear system affected by zero-mean Gaussian noise. Furthermore, the time for computational processing for both implementations is of the same order of magnitude.

## 1. Introduction

Recently, the use of diverse types of sensors and different strategies for information fusion has allowed important developments in key areas of robotic and artificial intelligence. Within these disciplines, a specific area of investigation is mobile robotics where the sensor-based localization problem is an important research topic. Localization of an autonomous mobile robot is the main concern of a navigation strategy since it is necessary to know precisely the actual position of the mobile robot in order to apply a control law or execute a desired task. In general, a navigation system requires a set of sensors and a fusion algorithm that integrates the sensors information to reliably estimate the pose of the mobile robot. One of the most commonly used sensors in wheeled mobile robots is odometers (dead reckoning). Unfortunately, these sensors are subjected to accumulated errors introduced by wheel slippage or other uncertainties

that may perturb the course of the robot. Therefore, odometric estimations need to be corrected by a complementary type of sensor. Reported works in autonomous robots present approaches where the odometry sensors information is complemented with different types of sensors such as ultrasonic sensor [1–3], LIDAR (Light Detection and Ranging) [4–8], digital cameras [9–12], magnetic field sensor [13], global position system (GPS) [7, 8, 14, 15], and Inertia Measurement Units (IMUs) [7, 15].

Among the different types of sensors there exist advantages and drawbacks depending on the general application of the mobile robots considered. GPS systems are low-cost systems and relatively easy to implement but have low accuracy and their use is not convenient for indoor environments. IMUs are relatively inexpensive, easy to implement, and efficient for outdoor and indoor conditions but are very sensitive to vibration-like noise and are not convenient for precise applications. LIDAR sensors have high accuracy and are

robust for indoor and outdoor applications, with acceptable performance in variable light conditions; however, LIDAR sensors are expensive and the data processing is complex and time consuming. Camera sensors are inexpensive and easy to implement and are with an important amount of tools for images processing and analysis. Although vision sensors are sensitive to light and weather conditions, the use of vision sensors for indoor-structured environments with controlled light conditions is very reliable.

When several sensors are implemented in a single intelligent system (e.g., a mobile robot), it becomes necessary to implement a strategy to fuse the data from every sensor in order to optimize information. To this end, combining sensor data is usually called sensors fusion. With respect to the localization problem, the specialized literature reported several fusion strategies [16]. These techniques can be classified in heuristics algorithms (e.g., genetic algorithm and fuzzy logic) [3], optimal algorithms (Kalman Filter and grid-based estimations) [15], and suboptimal algorithms. Real-world problems normally utilize suboptimal Bayesian filtering, such as approximated grid-based estimations, Extended or Unscented Kalman Filters [9, 12, 17–19], and particles filtering methods [1, 20]. Due to their ability to perform real-time processing and reliability, Kalman-based fusion techniques are implemented in many cases, under the assumption that the noise affecting the system is zero-mean and Gaussian. However, for the case of a robotic wheelchair such assumption is rather strict and is not always satisfied [21].

Wheelchairs, unlike specialized mobile robots, show many uncertainties related to their inexpensive construction, foldable structure, and nonholonomic characteristics of the wheels. Hence, nonlinear and non-Gaussian assumptions become important for these low-end vehicles, where pose uncertainty can be a consequence of differing wheel diameters, as well as gross misalignment, dynamic unbalances, or other problems due to daily use and factory defects. Thus, the filtering strategy is crucial in order to minimize all the uncertainties that are not considered in an ideal mathematical model.

Considering the special case where the mobile robot is a wheelchair intended to be used by a severely disabled person, the literature offers some good examples of fusions between different sensors using nonoptimal algorithms. In [22], the wheelchair is controlled by the user through special devices where a brain-machine interface control is proposed for semiautonomous drive. In [13], a line-follower-like wheelchair moves autonomously, updating its position through metal-artificial markers and RFID tags. In [5, 23], a vision-based autonomous navigation control using an EKF and artificial markers is described. In [1], an autonomous wheelchair is developed using odometry and ultrasonic sensor measurements to be fused using a PF.

In this work, a vision-based particle filter (PF) fusion algorithm is proposed and compared with a Kalman-based algorithm. Particle filters are able to work on the assumption that nonlinear systems are affected by non-Gaussian noise but can be computationally intensive. A novel implementation of a PF on sensor space, called camera space particle filter

(CSPF), is used to combine visual observations with sensed wheel rotations mapped onto a camera space through an observation function. A main contribution of this project is the novel strategy of the CSPF implementation. The CSPF fuses the data from odometry and vision sensors on camera space resulting in a precise update of the wheelchair's pose. The particles used by the CSPF are a set of odometry estimations each with random initial conditions. Every first estimated position is mapped into a camera space through an observation function. In this work, the PF is performed in the sensor space with every visual measurement. Using this strategy, the computational demand is reduced considerably since the filtering is applied to a set of horizontal pixel positions and a single marker observation (point of interest), avoiding the need for exhaustive image processing. The computational processing time of the implementation here presented is shown to be of the same order of magnitude as a typical Kalman Filter implementation.

## 2. Methods and Materials

In this work, only the nominal kinematic model is required to estimate the position of the wheelchair. If the kinematic model considers only non-holonomic constraints, real-world disturbances such as slipping and sliding of the wheels with respect to the ground (or any error from other sources) are not taken into account and therefore must be corrected. To update the wheelchair positions, the kinematic model is coupled with a pinhole camera model. Applying the solution here proposed, observations of passive artificial markers at known positions are used to estimate the wheelchair physical position through an observation function that maps from the planar physical space to camera space. Next, the kinematical model and the approach used to set up the observation function are first reviewed followed by the camera space particle filter algorithm description.

*2.1. Navigation Strategy.* In this approach a metric map is built from a training stage. Here, a person drives the wheelchair through a desired path where different estimated positions of the wheelchair are recorded. Based on the acquired metric map, the wheelchair moves autonomously following the instructions recorded in the training stage. For both stages, either when the wheelchair builds the metric map or when the wheelchair tracks the reference path, the wheelchair estimates its position based on the CSPF here proposed. This strategy is convenient for disabled users since it is not necessary to visit every place inside the work area as proposed in other approaches used in mobile robotics typically used for exploration such as Simultaneous Localization and Mapping (SLAM) [6].

*2.2. Kinematic Model.* The kinematic model used for controlling the wheelchair is a unicycle. The reference point  $X(X, Y, \phi)$  used to identify the position of the wheelchair in the physical space is assumed to be at the middle of the traction wheels axis; see Figure 1. The  $X$  and  $Y$  coordinates

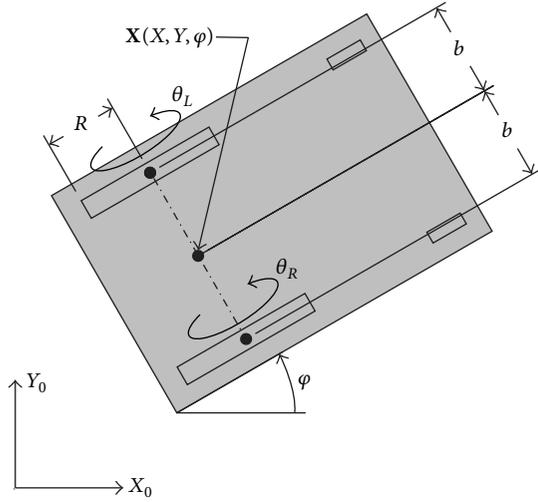


FIGURE 1: Wheelchair kinematic scheme model.

are considered with respect to an inertial fixed reference frame  $X_0 - Y_0$ .  $R$  is the radius of traction wheels and  $2b$  is the distance between the wheels. The wheelchair's orientation angle is represented by  $\varphi = \varphi(\alpha)$ , where  $\alpha$  defines the average between the right and left driving wheels' rotation angles  $\theta_R$  and  $\theta_L$ , respectively:

$$\alpha = \frac{\theta_R + \theta_L}{2}. \quad (1)$$

The position of  $\mathbf{X}(X, Y, \varphi)$  is obtained from integrating the following kinematic system of equations:

$$\begin{aligned} \frac{dX}{d\alpha} &= R \cos \varphi, \\ \frac{dY}{d\alpha} &= R \sin \varphi, \\ \frac{d\varphi}{d\alpha} &= \frac{R}{b} u; \end{aligned} \quad (2)$$

the variable  $u$  can be defined as a function of the differential forward rotations of the two driving wheels:

$$u = \frac{d\theta_R - d\theta_L}{d\theta_R + d\theta_L} \approx \frac{\Delta\theta_R - \Delta\theta_L}{\Delta\theta_R + \Delta\theta_L}; \quad (3)$$

the state of the system is defined by the wheelchair's pose  $\mathbf{X} \equiv [X, Y, \varphi]^T$ . In general, (2) can be expressed compactly as

$$\frac{d\mathbf{X}}{d\alpha} = \begin{bmatrix} \frac{dX}{d\alpha} & \frac{dY}{d\alpha} & \frac{d\varphi}{d\alpha} \end{bmatrix}^T = \mathbf{f}(X, Y, \varphi); \quad (4)$$

thus, (4) can be solved by odometry integration.

**2.3. Observation Function.** A vision function based on the pinhole camera model was defined using four vision

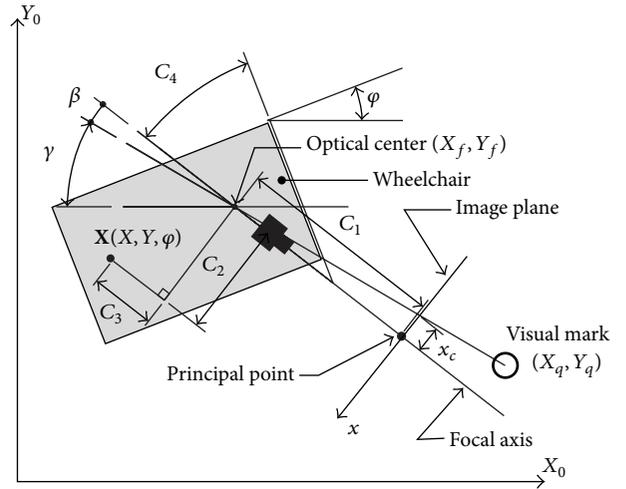


FIGURE 2: Description of the vision parameters used at the observations.

parameters (see Figure 2) where  $C_1$  represents the distance in pixels from the image plane to the optical center along the focal axis,  $C_2$  is the orthogonal distance in mm between the focal axis and point  $P$ ,  $C_3$  is the distance in mm between the optical center and point  $P$  parallel to focal axis, and  $C_4$  is the fix angle in radians between the wheels' axis and the focal axis.

The reference frame  $x - y$  is defined as the camera space coordinate system which has its origin at the principal point of the image plane, Figure 2.

Based on Figure 2, (5) is obtained, where  $x_c$  is negative since it is placed at the left side of the principal point:

$$\tan \beta = \frac{-x_c}{C_1}; \quad (5)$$

angle  $\beta$  is formed between the focal axis and the projected axis of the observed marker,  $\gamma$  is the angle between the  $X_0$  axis and the projected axis of the observed marker, and  $\varphi$  is the orientation angle of the wheelchair with respect to the global coordinate  $X_0$ ; see Figure 2. These angles are related to  $C_4$  as it is shown in

$$\beta = \frac{\pi}{2} - (\gamma + \varphi + C_4); \quad (6)$$

substituting (6) in (5) yields

$$x_c = -C_1 \tan \left[ \frac{\pi}{2} - (\gamma + \varphi + C_4) \right]; \quad (7)$$

after some trigonometric manipulation, the following equation is obtained:

$$x_c = -C_1 \frac{\cos(\varphi + C_4) - \sin(\varphi + C_4) \tan \gamma}{\sin(\varphi + C_4) + \cos(\varphi + C_4) \tan \gamma}. \quad (8)$$

From Figure 2, the following equation can be verified:

$$\tan \gamma = \frac{-(Y_q - Y_f)}{(X_q - X_f)}. \quad (9)$$

Thus, substituting (9) into (8) produces

$$x_c = -C_1 \frac{(X_q - X_f) \cos(\varphi + C_4) + (Y_q - Y_f) \sin(\varphi + C_4)}{(X_q - X_f) \sin(\varphi + C_4) - (Y_q - Y_f) \cos(\varphi + C_4)}, \quad (10)$$

$$x_c = -C_1 \frac{(X_{q,v} - X) \cos(\varphi + C_4) + (Y_{q,v} - Y) \sin(\varphi + C_4) - C_3 \sin[2(\varphi + C_4)] - C_2}{(X_{q,v} - X) \sin(\varphi + C_4) - (Y_{q,v} - Y) \cos(\varphi + C_4) + C_3 \cos[2(\varphi + C_4)]}, \quad (12)$$

where  $X_{q,v}, Y_{q,v}$  are known coordinates of a given  $v$  visual marker. This function maps the physical space coordinates into the camera space. The variable  $x_c$  is defined as the projected distance along  $x$  between the focal axis and the visual marker's centroid on camera space coordinate system. Thus, (12) can be defined as the observation function  $\mathbf{h}$ :

$$\mathbf{x} = [x_c] = \mathbf{h}(X, Y, \varphi). \quad (13)$$

This observation function is henceforth used to map the planar coordinates of the wheelchair  $(X, Y, \varphi)$  in the physical space to the  $(x_c, y_c)$  camera space coordinate system. It is noteworthy to observe that  $x_c$ , defined within the image plane location, is the only variable used in this study, since the marker's vertical location of the centroid on image plane is set to be constant. Thus,  $y_c$  is irrelevant for the wheelchair position estimation as the system (including the cameras) is constrained to travel on a plane.

To estimate the vision parameters values  $C_1, C_2, C_3$ , and  $C_4$  the wheelchair is taken to different positions across the planar physical space where visual markers can be observed. Whenever a visual marker is detected, both the visual marker position on camera space and its position on physical space are associated with the wheelchair position on physical space; this information is saved in a "training database." Based on this information and using (12) a least squares minimization via a Marquardt method process is performed to compute the vision parameters.

**2.4. Camera Space Particle Filter.** A particle filter is a statistical, exhaustive search approach to estimation that often works well for problems that are difficult for conventional Extended Kalman Filter (i.e., systems that are highly nonlinear or affected by non-Gaussian noise) [24]. The main idea on the PF is to represent a posterior density function by a set of random samples with associated weights and compute estimates based on these samples and weights [16].

where  $(X_q, Y_q)$  and  $(X_f, Y_f)$  are, respectively, the coordinates of the observed marker and the position of the focus of the camera with respect to the fixed frame of reference  $(X, Y)$ .

The coordinates  $(X_f, Y_f)$  are described by

$$\begin{aligned} X_f &= X + C_3 \sin(\varphi + C_4) + C_2 \cos(\varphi + C_4), \\ Y_f &= Y + C_3 \cos(\varphi + C_4) + C_2 \sin(\varphi + C_4). \end{aligned} \quad (11)$$

Using these values into (10) yields the observation function

To define the problem of tracking, consider the evolution of the state sequence of a target given by the system and measurement equations:

$$\begin{aligned} \mathbf{X}_k &= \mathbf{f}(\mathbf{X}_{k-1}) + \mathbf{V}_{k-1}, \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{X}_k) + \mathbf{n}_k, \end{aligned} \quad (14)$$

where  $\mathbf{f}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$  is a nonlinear function of the state  $\mathbf{X}_{k-1}$ ,  $\mathbf{h}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_z}$  is a nonlinear function related with the measurement process, and  $k$  is the  $\alpha$  step index. Furthermore,  $\mathbf{V}_{k-1}$  and  $\mathbf{n}_{k-1}$  are system and measurement noises vectors, respectively; both are independent and identically distributed (iid).

The algorithm proposed in this work was based on the PF approach described by [24], which in turn is based on the Monte Carlo method; this algorithm is summarized as follows.

- (1) Assuming that the probability density function (pdf) of the initial state  $p(\mathbf{X}_0)$  is known,  $N_s$  initial states are randomly generated based on  $p(\mathbf{X}_0)$ ; these states are denoted by  $\{\mathbf{X}_{0,i}^+, i = 0, \dots, N_s\}$  and are uniformly distributed over an initial range  $\mathbf{W}$ , that is,  $\mathbf{X}_{0,i}^+ \sim \mathbb{U}[\mathbf{X}_0 - \mathbf{W}, \mathbf{X}_0 + \mathbf{W}]$ , where  $\mathbf{W}$  is tuned experimentally;  $\mathbb{U}[a, b]$  is a closed set of random numbers uniformly distributed with values from  $a$  up to  $b$ , and the parameter  $N_s$  (number of particles) is chosen as a trade-off between computational effort and estimation accuracy.
- (2) For each state update, the algorithm is as follows.

- (a) Perform the  $\alpha$  propagation step to obtain an a priori state  $\mathbf{X}_{k,i}^-$ , through

$$\mathbf{X}_{k,i}^- = \mathbf{f}(\mathbf{X}_{k-1,i}^+) + \mathbf{V}_{k-1}, \quad i = 1, \dots, N_s. \quad (15)$$

- (b) A set of particles on camera space is determined using the observation equation

$$\mathbf{x}_{k,i}^- = \mathbf{h}(\mathbf{X}_{k,i}^-). \quad (16)$$

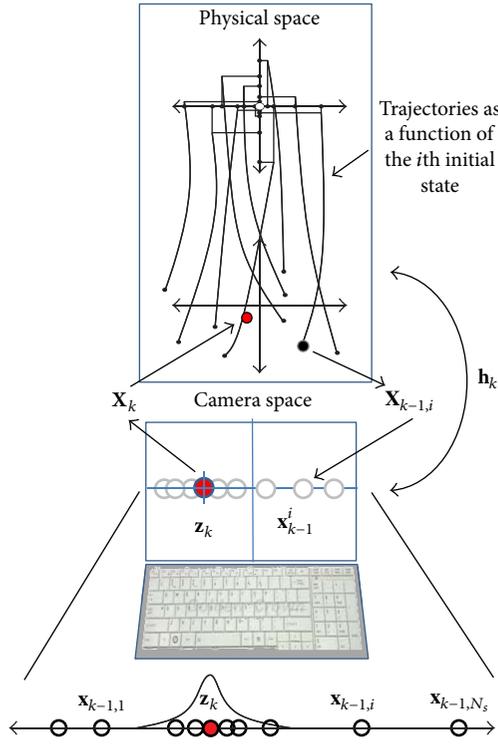


FIGURE 3: Scheme of PF on camera space.

- (c) The weight  $\{w_{k,i}, i = 1, \dots, N_s\}$  for each particle  $\mathbf{x}_{k,i}^-$  is conditioned on the measurement  $\mathbf{z}_k$  through (17). For every particle, the weight  $w_{k,i}$  is the maximum at the central pixel of the visual marker and decreased following the tail of a Gaussian function as the particle is farther away from the central pixel of the visual marker position observed on the camera space; see Figure 3:

$$w_{k,i} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2 \cdot \sigma^2} [\mathbf{z}_k - \mathbf{x}_{k,i}^-]^T [\mathbf{z}_k - \mathbf{x}_{k,i}^-]\right); \quad (17)$$

in (17)  $\sigma$  is the covariance of the measurement noise tuned experimentally.

- (d) The weights obtained in the previous step are normalized as follows:

$$w_{k,i} = \frac{w_{k,i}}{\sum_{j=1}^{N_s} w_{k,j}}; \quad (18)$$

the sum of all relative weights is equal to one.

- (e) Next a resampling step is needed to generate a set of posterior particles  $\mathbf{x}_{k,i}^+$  on the basis of the relative weights  $w_{k,i}$  as follows. For  $i = 1, \dots, N_s$  the following three steps were performed.

- (i) A random number  $\tau \sim \mathbb{U}[0, 1]$  is generated.
- (ii) Accumulating the weights  $w_{k,i}$  into a sum, one at a time, until the cumulative sum is greater than  $\tau$ , that is,  $\sum_{m=1}^{j-1} w_{k,m} < \tau$  but

$\sum_{m=1}^j w_{k,m} > \tau$ , the new particle  $\mathbf{x}_{k,i}^+$  is then set to be equal to the old particle  $\mathbf{x}_{k,j}^-$ ; the resampling step is summarized as

$$\mathbf{x}_{k,i}^+ = \mathbf{x}_{k,j}^- \quad \text{with probability } w_{k,j}; \quad (19)$$

$\mathbf{X}_{k,i}^+$  is related to its state  $\mathbf{X}_{k,i}^+$ , so that  $\mathbf{X}_{k,i}^+$  takes the old state value  $\mathbf{X}_{k,j}^-$ ; the average value of the set  $\{\mathbf{x}_{k,i}^+, i = 1, 2, \dots, N_s\}$  becomes the state estimation  $\mathbf{X}_k$ .

- (iii) After the PF update, a new set of random states is set, so impoverishment is avoided. The new states are uniformly distributed in the screen window  $[\mathbf{X}_k - \mathbf{W}, \mathbf{X}_k + \mathbf{W}]$  about  $\mathbf{X}_k$ , which are  $\mathbf{X}_{k-1,i}^+$  in the next iteration.

Filtering on camera space implies using a PF in a one-dimensional set; this set is formed by the centroid of the observed visual markers ( $\mathbf{z}_k$ ) and all those virtual markers coming from observation equation (12) that use all different possible positions estimated through odometer data before PF correction.

**2.5. Testing Platform Description.** A low-cost foldable electrical wheelchair model P9000 XDT from INVACARE was used for this experiment. This wheelchair is usually controlled through a joystick which allows displacement and velocity control.

The P9000 XDT wheelchair has two DC brush motors which were fitted with encoders at each motor and connected to a Galil Motion Control Board DMC-4040. The encoders used for this implementation were two-channel incremental encoders from Avago Technologies, model HEDS-5600 #B06. As vision sensors, two uEye cameras,  $640 \times 480$  pixels, were installed on the wheelchair. An emergency stop button was added. This equipment is depicted in Figure 4.

A computer program was developed using Microsoft Visual C++ 2010 and open access libraries OpenCV 2.3, running in operating system Windows 7 Home Premium edition on an on-board Toshiba laptop Core i5 processor at 2.5 GHz and 4 GB RAM memory.

### 3. Results and Discussion

**3.1. Experiment Description.** To test the algorithm described in Section 2.3, a set of experiments was developed. For these experiments, the path to be followed by the wheelchair and its initial position were defined. Along this path  $N_c$  visual markers (concentric circles) were placed at known positions  $(X_{q,v}, Y_{q,v})$ , shown in Figure 5.

After the markers' positions were established and a desired path was chosen, a training stage was performed in order to track automatically the desired path. The wheelchair is trained by a human guide that drives the wheelchair along a path by each position where visual markers can be detected by the cameras. Using odometry, the wheelchair position is estimated and mapped into camera space. Thus, the differences between the odometer estimation in camera



FIGURE 4: Experimental wheelchair and sensors placement.



FIGURE 5: Artificial visual markers and wheelchair at initial position.

space and the observed marker position on camera space are used to update the wheelchair's actual position according to the fusion filter applied (e.g., CSPF and EKF). This position (i.e., the state of the system) is saved in a database.

Based on the training information, the wheelchair moves to each of the saved states. In such locations, the vision system scans for a marker related to the current state. Based on a filtering strategy, the acquisition of the marker allows the updating of the wheelchair's pose. After the pose is updated, it is possible to perform a control strategy to arrive to the next position.

Two different experiments were implemented to validate the system. The first test consisted of a "Straight line" segment 7.5 m long (Figure 6); the second experiment was an "L-path" trajectory (Figure 7). In these two experiments, the wheelchair followed a trained trajectory (i.e., a reference path) using both CSPF and EKF. Finally, a third experiment to test the CSPF in complex maneuvers was implemented; see Figure 8. This experiment is called "Office-path" where the wheelchair goes from a desk located in an office to a second desk in a different office passing through a narrow door following a reference path.

**3.2. Experimental Results.** The physical position automatically tracked by the wheelchair was measured using

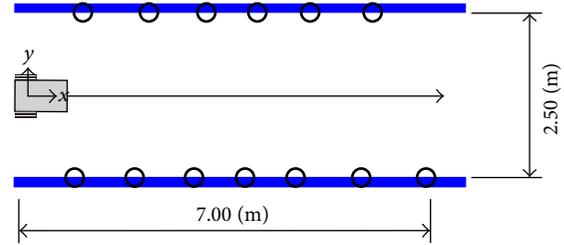


FIGURE 6: "Straight line" experiment scheme.

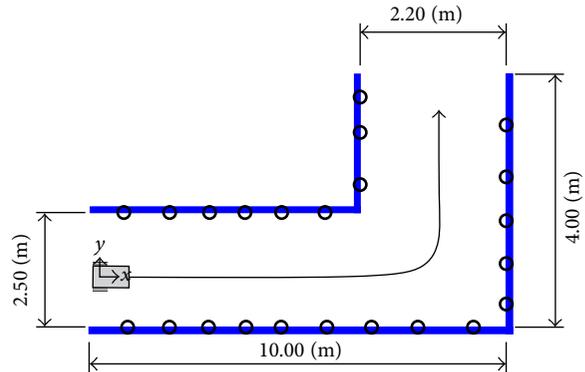


FIGURE 7: "L-path" experiment scheme.

a measuring tape with uncertainty of about 0.01 m. Results for the "Straight line," "L-path," and "Office-path" experiments are displayed in Figures 9, 10, and 11, respectively. For these experiments, the number of particles was set experimentally to  $N_s = 500$  and the width range for the initial random positions in the PF implementation was  $\mathbf{W} = [0.02 \text{ m}, 0.02 \text{ m}, 0.01 \text{ rad}]^T$ .

**3.2.1. "Straight Line" Experiment.** For the "Straight line" experiment, the wheelchair moves through a 7.5 m long segment (Figure 6). During the training stage, the wheelchair was driven following a straight line painted on the floor, allowing the human guide to drive the system precisely.

Results from the "Straight line" experiment are shown in Figure 9. This graph shows four series of data; the first one is the reference path recorded in the training stage, and the rest of the series are the measurement of the automatically tracked positions during the task performed using different types of filters. First, in the "No-filter" series the wheelchair follows the references path based only on the recorded information and the odometry sensors without filtering update (i.e., correction). This task produced a maximum error at the end of the path equal to 0.3 m. This high magnitude of error arises because the initial conditions are not set accurately enough and because other uncertainties cannot be accounted for into the kinematic model. Thus, it is necessary to have additional information (i.e., camera observations) to detect deviations from the reference path. The task where an EKF fuses odometry and camera information shows deviations of less than 0.05 m in Y direction from the reference path. Finally, the task where a CSPF is used as the filter shows a

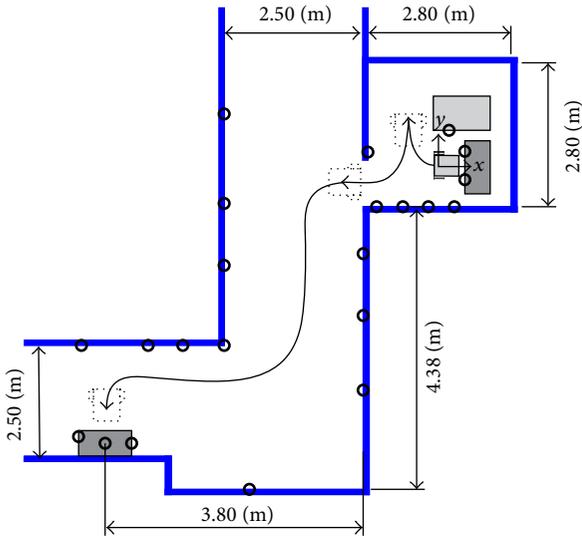


FIGURE 8: "Office-path" experiment schema.

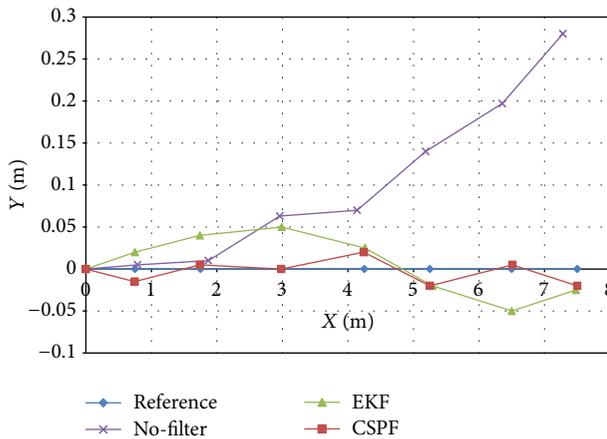


FIGURE 9: Positions measurement of the reference path and the wheelchair tracked positions using different filters for the "Straight line" experiment.

deviation of about 0.03 m in the Y direction. Figure 9 shows that our implementation of CSPF yields better performance in terms of transversal deviation compared to the EKF.

It is noteworthy that the CSPF is a filter that minimizes non-Gaussian noise surely present in a foldable wheelchair system, due to wheels differences, unbalances, and so forth. These sources of noise skew the probability density function, so it is harder for the EKF to correct the errors affecting the wheelchair pose.

**3.2.2. "L-Path" Experiment.** For the "L-path" experiment, the reference path is an L-like path which consists of a straight segment of approximately 8 m connected through a curve to a second 2.5 m perpendicular segment; this trajectory is about 11 m long; see Figure 7.

The results for "L-path" experiment are shown in Figure 10. Here, the series with no-filter clearly showed a large

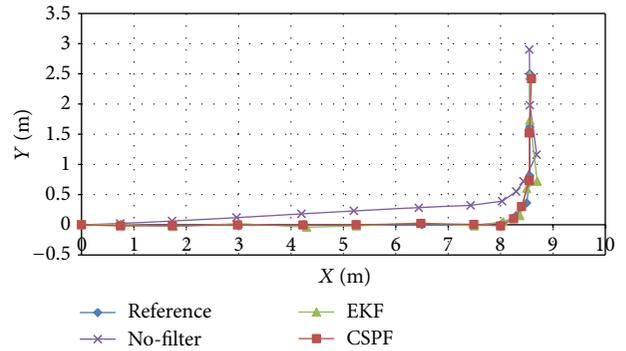


FIGURE 10: Positions measurement of the reference path and the wheelchair tracked positions using different filters for the "L-path" experiment.

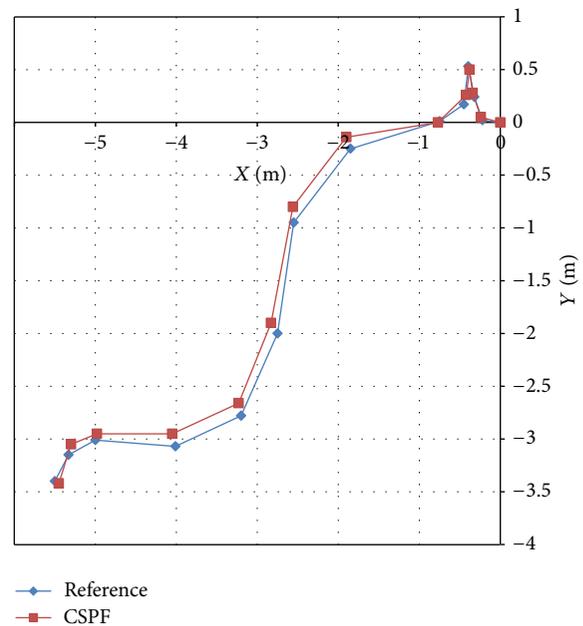


FIGURE 11: Estimations and measurements results using PF at "office" experiment.

deviation from the reference path. In the first segment, the wheelchair deviated about 0.4 m along the Y direction from the reference path. When approaching the 90 degrees, the wheelchair did not turn sharply. In this case, the transversal error compensates by increasing the longitudinal deviation. In the second segment, although the error in the transversal direction does not seem significant, the wheelchair overreaches the final position (i.e., Y direction) of about 0.5 m.

During the first segment, when the task is performed via EKF, the performance seems very similar to the task performed via CSPF. At the beginning of the second segment, the EKF shows a deviation from the reference path of about 0.25 m. At the point where the curve connects with the second segment, the deviation from the reference path obtained using the CSPF-based tracking is always smaller than the one from the task performed with the EKF-based corrections.

TABLE 1: RMS error in m of positions reached using different filter techniques against a reference.

Experiment	Coordinate	RMS error		
		No-filter	EKF	CSPF
Line	X	0.113	0.017	0.009
	Y	0.135	0.033	0.013
L-path	X	0.045	0.042	0.033
	Y	0.268	0.076	0.048
Office-path	X	—	—	0.033
	Y	—	—	0.079

3.2.3. “Office-Path” Experiment. Finally, the CSPF was tested in a new trajectory where the wheelchair is tasked to go from one office to another. Furthermore, the wheelchair is placed at an office desk and is required to automatically drive to a different desk. In this experiment, the wheelchair needs to perform various complex maneuvers such as moving backward, turning, and moving forward. This trajectory includes passing through a doorway (0.9 m width), making a left turn followed by a right turn, and finally taking the wheelchair into a second desk. This trip is approximately 10 m long, with a higher complexity level than the previous two experiments.

Due to the backward movement of the wheelchair needed at the beginning of the trajectory in the “Office-path” tracking via EKF was not considered. Indeed, during backward movements, the EKF requires a different algorithm where it is necessary to subtract the covariance matrix error instead of adding it [23]. This difference does not apply for CSPF since the variation is directly considered from the odometry information that produces the PDF filtered by the CSPF, so it is suitable either for forward or for backward maneuvers.

Figure 11 shows the results from the “Office-path” experiment where the wheelchair is able to follow the reference path with no collisions. It is noteworthy that when the experiment was performed without any filter the wheelchair was not able to come out from the office due to repeated impacts with the doorway. This trajectory is not shown.

3.2.4. RMS Analysis. In order to facilitate the performance comparison between CSPF, EKF, and no-filter the root mean square (RMS) error from the reference path is proposed. The root mean square error is defined as the squared root of the mean squared difference between the position reached using a filter technique and the reference path:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{X}_i^{\text{ref}} - \mathbf{X}_i^{\text{filter}})^2}, \quad (20)$$

where  $N$  is the number of corrections made during a trip and  $\mathbf{X}_i^{\text{ref}} = [X_i^{\text{ref}}, Y_i^{\text{ref}}]^T$  is the physical measurement of the  $X$  and  $Y$  coordinates of the reference path at a moment in which an  $i$  correction is performed.  $\mathbf{X}_i^{\text{filter}} = [X_i^{\text{filter}}, Y_i^{\text{filter}}]^T$  is the physical measurement of the  $X$  and  $Y$  coordinates at the tracked position when the  $i$ th correction is performed using a certain type of filter technique.

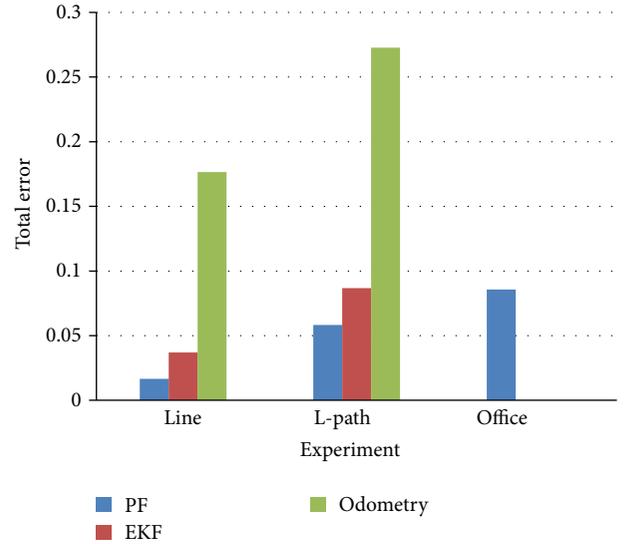


FIGURE 12: Total error in m for different experiments by type of filtering technique.

Table 1 shows the RMS error analysis for both the  $X$  and  $Y$  coordinates. For the “Straight line” and “L-path” the deviation increases significantly for both  $X$  and  $Y$  coordinates when No-filter is used compared to EKF and CSPF. This deviation is expected because the control is open-loop and there is no feed-back pathway to measure any deviation from the reference path. Therefore, the system is very sensitive to variations of initial conditions and uncertainties. In Table 1, the “Office-path” experiment does not show values for either the No-filter or EKF conditions. In the first condition, it was not possible to complete the task because the wheelchair usually impact the doorway. In the second case, a change in the test algorithm would have been necessary in order to perform the task. When both the EKF and CSPF implementations were possible, both filters performed well with better tracking when the CSPF was used.

In Figure 12 the values for the total error ( $\sqrt{\text{RMS}_X^2 + \text{RMS}_Y^2}$ ) are described. The largest total error is obtained along the “L-path” experiment when No-filter technique is utilized. The error is 0.272 m which is about 50% of the width of the wheelchair (i.e., 0.6 m). In Figure 12, EKF and CSPF can be compared and used to perform the “Line” and “L-path” task. During the “Line” task, the EKF

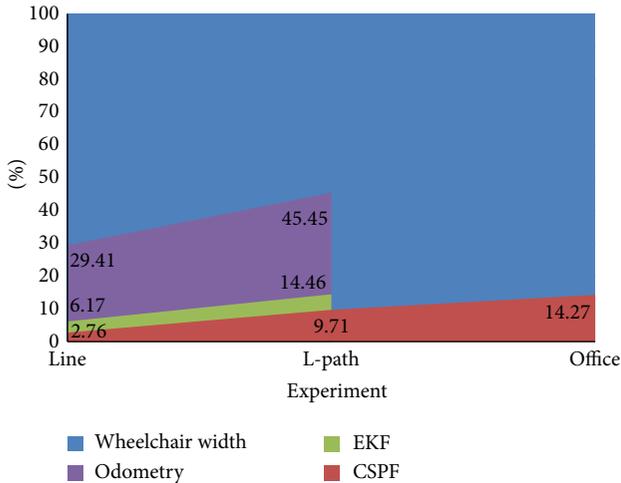


FIGURE 13: Total deviation compared with the wheelchair width.

yields a total error of 0.037 m (6.0% of the wheelchair width), while the CSPF implementation yields 0.016 m (2.8% of the wheelchair width). Along the L-path series depicted in Figure 12 due to the increased complexity, the total error also increases to 0.086 m for EKF and 0.058 m for CSPF (14.5% and 9% of wheelchair width, resp.). These values are safe values considering halls with more than 2 m of clearance. Even when passing a doorway, these values could be tolerable since the differences between the wheelchair width and the door clearance are about 3 m (i.e., 50% of the wheelchair). The “Office-experiment” task shows an error of 0.085 m (14.2% of the wheelchair width). This error still guarantees a good safety margin to avoid collision, considering that the reference path passes closely to the middle of the doorway ( $\pm 0.1$  m about the center); see Figure 13.

**3.2.5. Processing Time.** In this approach, the increase in computational cost between the EKF and CSPF implementation is not very critical. This is a consequence of implementing the particle filter algorithm in sensor space. Furthermore, the observation function implemented in this work allows for mapping a tridimensional to a one-dimensional space saving an important amount of computational effort.

Figure 14 depicts the processing time of 4 CSPF implementations with different amount of particles, alongside the processing time of EKF implementation. For the CSPF to have a reliable performance 500 particles were necessary, thus, increasing the processing time of only 10 ms with respect to the EKF implementation. For this application, increasing the processing time of 10 ms per cycle does not compromise the real-time performance. On the other hand, even though the CSPF implementation requires a slightly longer processing, an improvement in the accuracy is achieved. Furthermore, the CSPF algorithm does not need special considerations for forward or backward displacements, using a simpler implementation. Finally, the CSPF is limited by neither linearity nor Gaussianity assumptions as it is the case for EKF implementations.

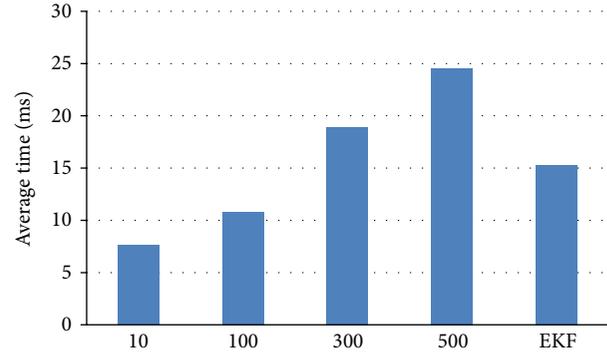


FIGURE 14: Comparison of processing time between EKF and PF with different particles amount.

## 4. Conclusions

The theoretical development and experimental implementation of a vision-based sensing technique for the robust and precise localization of an automatically guided wheelchair were presented. This system is designed to support individuals with severe disabilities. Such application requires accurate localization and control.

The developed algorithm consists of an original implementation of particle filtering on camera space (CSPF). This work takes advantage of the intrinsic characteristics of the particle filter (i.e., ability to deal with non-zero-mean and non-Gaussian uncertainties where the probability density function is not bound to Gaussian type). Moreover, the implemented sensor-based method avoids time consuming state estimations and inherent model errors.

A successful experimental testing showed the feasibility of the proposed approach. The system was capable of following complex trajectories in indoor-structured environments where tight tolerances are required such as when passing through a doorway or docking into a desk. The proposed algorithm, based on experiments, proved to be robust to the uncertainties of the system.

The system performs a precise position estimation based on odometry and vision information through a novel CSPF implementation that fuses the sensors information. In the fusion process, the inaccurate odometry information (dead reckoning) is taken to camera space through an observation function where, using the Monte Carlo method, a set of most likely positions is produced. Finally an estimation of the system’s position is obtained through the average of the resulting set of positions.

The CSPF is not limited to linear or Gaussian assumptions unlike the common case of Kalman-based filters. This is an advantage when using the filter on systems subjected to biased uncertainties as in the case of a foldable wheelchair.

The robotic wheelchair system implemented in this work is able to follow a variety of reference paths. Three experiments called “Straight line,” “L-path,” and “Office-path” were performed. In order to validate the system, the fusion of the sensors information using CSPF was compared with No-filter

and EKF implementations in two experiments: “Straight line” and “L-path.” For both experiments when No-filter was used the performance decreased significantly deeming the system useless for the application. When the CSPF and EKF were compared, results in both experiments showed that CSPF had better performance. This may be due to the capability of the CSPF to perform using a distribution of uncertainties with skew shape of the PDF. On the other hand, EKF does not perform as well when the distribution of uncertainty is not Gaussian. After the system was validated against the EKF implementation, the system was tested in a significantly more complex trajectory, the “Office-path.” For this path the RMS error was slightly larger than for the other cases due to the added complexity of the trajectory but results were still within acceptable and safe values for the users.

As a mean of comparison to other studies, [25] shows a study where using fixed cameras several filtering techniques are applying for the localization of mobile robots playing soccer. Using EKF the localization RMS error was about 0.061 m. On the other hand, using a PF based algorithm this error decreases to 0.030 m, this being the most precise result reported. The robots utilized in [25] are 3-wheeled robots that move in an area of  $3 \times 5$  m. These robots have in general few sources of noise unlike the wheelchair used in this work. The shortest computational time performance was obtained using an EKF with an average processing time of about 1.3 ms. The time used in their PF implementation was 287.3 ms, which is 221 times the EKF processing time. Considering these results the present work shows an important contribution since it is able to localize precisely a noisy system, as it is the case for a foldable wheelchair, still maintaining the processing time of the same order of magnitude compared to an EKF implementation without linear or Gaussian limitations.

It is noteworthy to reiterate that in a CSPF the process which fuses the odometry information (dead reckoning), with the information from the digital cameras, is implemented within sensor space (i.e., camera space). The observation function is one-dimensional, reducing the computational burden and allowing a real-time implementation. The CSPF is a Monte Carlo-based algorithm that is not limited to linear or Gaussian assumptions unlike common implementations based upon Kalman Filters. This advantage is especially important in the case of a foldable wheelchair where the system is noisy and uncertainties are difficult to model.

To summarize, the advantages of the developed approach are numerous. The system developed in this work performs a precise and robust position estimation based on the combination of odometry and camera sensor space. An innovative aspect of the technique is avoiding the assumption of a linear model corrupted by Gaussian noise, assumptions that, instead, are common practice for Kalman-based filters. Our method reduces the computational burden typically associated with particle filter approaches allowing real-time implementations permitting the capability to follow complex paths. Additionally, new paths to be followed by the wheelchair can be easily set up during a training stage which is convenient for impaired users.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the Universidad Autonoma de Zacatecas and the Universidad Autonoma de San Luis Potosi and in part by CONACYT under Grant 213987.

## References

- [1] R. Grasse, Y. Morère, and A. Pruski, “Assisted navigation for persons with reduced mobility: path recognition through particle filtering (condensation algorithm),” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 60, no. 1, pp. 19–57, 2010.
- [2] S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, and J. Borenstein, “The navchair assistive wheelchair navigation system,” *IEEE Transactions on Rehabilitation Engineering*, vol. 7, no. 4, pp. 443–451, 1999.
- [3] Y. L. Ip, A. B. Rad, Y. K. Wong, Y. Liu, and X. M. Ren, “A localization algorithm for autonomous mobile robots via a fuzzy tuned extended Kalman filter,” *Advanced Robotics*, vol. 24, no. 1-2, pp. 179–206, 2010.
- [4] C. Sprunk, G. D. Tipaldi, A. Cherubini, and W. Burgard, “Lidar-based teach-and-repeat of mobile robot trajectories,” in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems: New Horizon (IROS '13)*, pp. 3144–3149, Tokyo, Japan, November 2013.
- [5] M. Pinto, H. Sobreira, A. P. Moreira, H. Mendonça, and A. Matos, “Self-localisation of indoor mobile robots using multi-hypotheses and a matching algorithm,” *Mechatronics*, vol. 23, no. 6, pp. 727–737, 2013.
- [6] A. Rusdinar, J. Kim, and S. Kim, “Error pose correction of mobile robot for SLAM problem using laser range finder based on particle filter,” in *Proceedings of the International Conference on Control, Automation and Systems*, Gyeonggi-do, Republic of Korea, 2010.
- [7] H. M. La, R. S. Lim, B. B. Basily et al., “Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1655–1664, 2013.
- [8] S. A. Hiremath, G. W. A. M. van der Heijden, F. K. van Evert, A. Stein, and C. J. F. Ter Braak, “Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter,” *Computers and Electronics in Agriculture*, vol. 100, pp. 41–50, 2014.
- [9] G. Del Castillo, S. Skaar, A. Cárdenas, and L. Fehr, “A sonar approach to obstacle detection for a vision-based autonomous wheelchair,” *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 967–981, 2006.
- [10] M.-H. Li, B.-R. Hong, Z.-S. Cai, S.-H. Piao, and Q.-C. Huang, “Novel indoor mobile robot navigation using monocular vision,” *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 485–497, 2008.
- [11] B. Romero, A. Camacho, J. Varona, C. Delgado, and R. Velázquez, “A low-cost electric power wheelchair with manual and vision-based control systems,” in *Proceedings of the IEEE AFRICON*, pp. 1–6, Nairobi, Kenya, September 2009.

- [12] Z. Jia, A. Balasuriya, and S. Challa, "Vision based data fusion for autonomous vehicles target tracking using interacting multiple dynamic models," *Computer Vision and Image Understanding*, vol. 109, no. 1, pp. 1–21, 2008.
- [13] C. de la Cruz, W. C. Celeste, and T. F. Bastos, "A robust navigation system for robotic wheelchairs," *Control Engineering Practice*, vol. 19, no. 6, pp. 575–590, 2011.
- [14] Y.-H. Wu, B.-Y. Lu, H.-Y. Chen et al., "The development of M3S-Based GPS navchair and tele-monitor system," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 4052–4055, Shanghai, China, September 2005.
- [15] F. Aghili and A. Salerno, "Driftless 3-D attitude determination and positioning of mobile robots by integration of IMU with two RTK GPSs," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 21–31, 2013.
- [16] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proceedings, Part F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.
- [17] E. T. Baumgartner, *An autonomous vision-based mobile robot [Ph.D. thesis]*, University of Notre Dame, Notre Dame, France, 1992.
- [18] J. A. C. Galindo, *A vision-guidance strategy to control hybrid holonomic/nonholonomic robots [Ph.D. thesis]*, University of Notre Dame, Notre Dame, France, 2003.
- [19] K. Jung, J. Kim, E. Jung, and S. Kim, "Positioning accuracy improvement of laser navigation using UKF and FIS," *Robotics and Autonomous Systems*, vol. 62, no. 9, pp. 1241–1247, 2014.
- [20] C. Pozna, R.-E. Precup, and P. Földesi, "A novel pose estimation algorithm for robotic navigation," *Robotics and Autonomous Systems*, vol. 63, no. 1, pp. 10–21, 2015.
- [21] G. Del Castillo, S. B. Skaar, and L. Fehr, "Extending teach and repeat to pivoting wheelchairs," *Systemics, Cybernetics and Informatics*, vol. 1, no. 1, pp. 55–62, 2003.
- [22] X. Perrin, R. Chavarriaga, F. Colas, R. Siegwart, and J. D. R. Millán, "Brain-coupled interaction for semi-autonomous navigation of an assistive robot," *Robotics and Autonomous Systems*, vol. 58, no. 12, pp. 1246–1255, 2010.
- [23] G. del Castillo, *Autonomous, vision-based, pivoting wheelchair with obstacle [Ph.D. thesis]*, University of Notre Dame, Notre Dame, France, 2004.
- [24] D. Simon, *Optimal State Estimation, Kalman, H<sub>∞</sub>, and Nonlinear Approaches*, John Wiley & Sons, Hoboken, NJ, USA, 2006.
- [25] J. L. Peralta-Cabezas, M. Torres-Torriti, and M. Guarini-Hermann, "A comparison of Bayesian prediction techniques for mobile robot trajectory tracking," *Robotica*, vol. 26, no. 5, pp. 571–585, 2008.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

