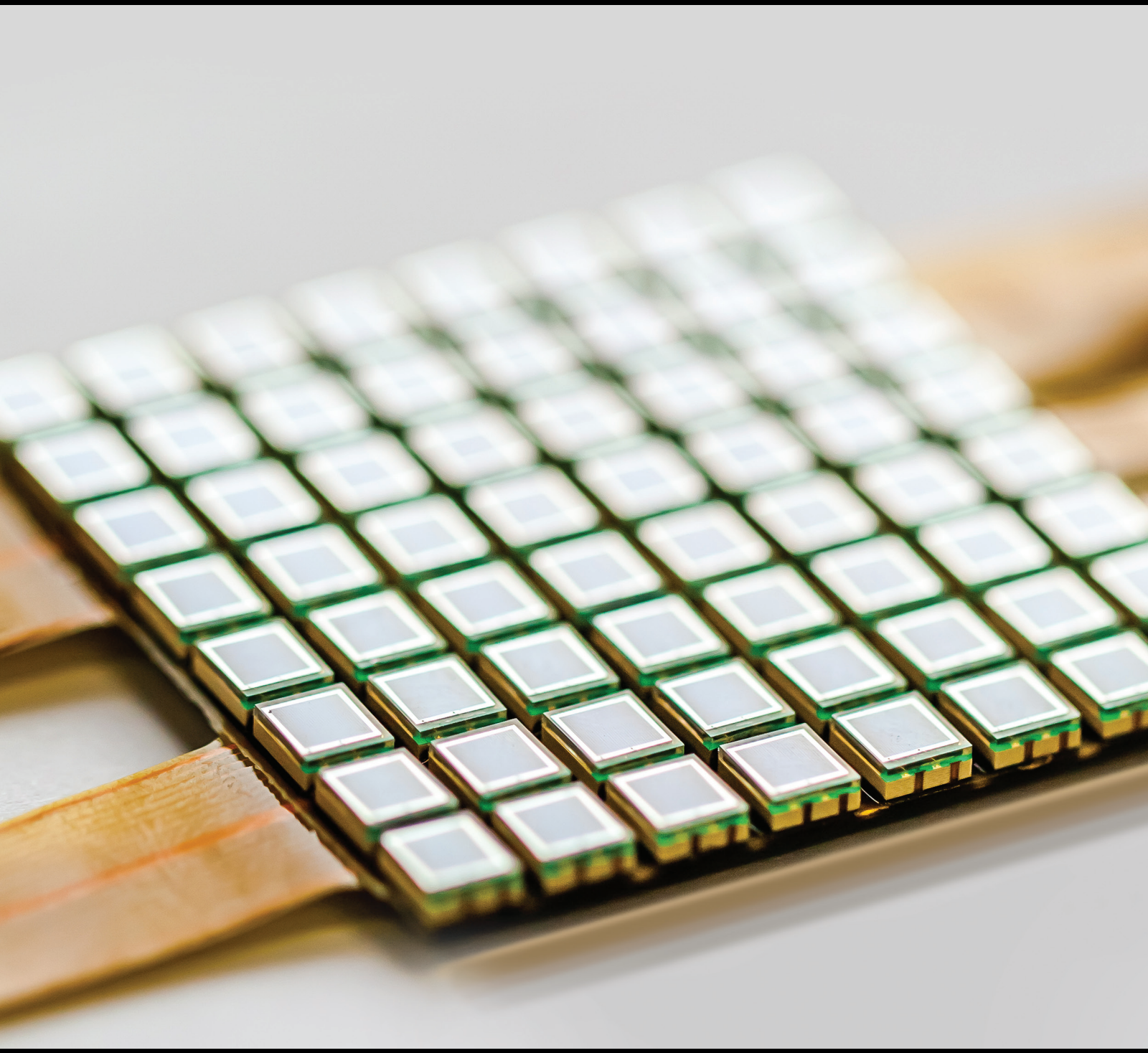
Journal of Sensors

Sensors for Robotics 2015

Guest Editors: Aiguo Song, Guangming Song, Daniela Constantinescu, Lei Wang, and Quanjun Song



Sensors for Robotics 2015

Journal of Sensors

Sensors for Robotics 2015

Guest Editors: Aiguo Song, Guangming Song,
Daniela Constantinescu, Lei Wang, and Quanjun Song

Editorial Board

Contents

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 412626, 2 pages
http://dx.doi.org/10.1155/2015/412626

Editorial

Sensors for Robotics 2015

Aiguo Song,[1] Guangming Song,[1] Daniela Constantinescu,[2] Lei Wang,[3] and Quanjun Song[4]

[1] *Robot Sensor and Control Lab, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China*
[2] *Department of Mechanical Engineering, University of Victoria, P.O. Box 3055, Victoria, BC, Canada STN CSC*
[3] *School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia*
[4] *Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China*

Correspondence should be addressed to Aiguo Song; a.g.song@seu.edu.cn

Received 1 July 2015; Accepted 6 July 2015

Robot is currently one of the exciting and fast developing technologies changing the life of human being. It has been widely applied in lots of areas such as industry, agriculture, medicine, transportation, social service, military, space exploration, and undersea exploiting. Increasing attention by robot researchers has been paid to the robot sensor, as a key component of the robot. During the last decade, much effort has been done to develop robot sensors for robot perception, robot control, autonomous robot, human-robot interaction, and so forth. In spite of the large and increasing interest and promising applications, robot sensor design is a significant challenging, which is involved in not only sensor materials, structure design, manufacturing process, and calibration technique, but also signal processing, data fusion, and pattern recognition. For instance, remarkable examples of tactile sensors and systems have been proposed; however, their ability to address specific applications and their extension to other fields such as medical instrumentation, prosthetic devices, and biomechanics test is questionable.

This special issue aims at exhibiting the latest research achievements, ideas, and advances in robot sensors. The special issue summarizes the most recent developments in the field of sensors for robotics. The theme of 2015 special issue focuses on the robot force and tactile sensor, robot sensor fusion, and robot sensor applications.

Force and tactile sensors are absolutely necessary elements for robot when interacting with environment. The paper by A. Almassri et al. surveys the state-of-the-art in variety force sensors for designing and application of robotic hand. This paper introduces the different techniques for measuring force or interface pressures. These techniques include load cells, pressure indicating film, and tactile pressure system. Similarly, a review on industry pressure sensing that involves the pick and place applications and algorithm control is also highlighted. The paper also discusses the MEMs sensor technology and different types of sensors. At last, it discusses the piezoresistive flexiforce sensor. Flexiforce sensor has a good substrate material, which is a polymer that enhances the force sensing and improves the performance of force, linearity, hysteresis, drift, and temperature sensitivity compared to any other thin film. Furthermore, it is flexible and ultrathin enough so that it can be widely used as robot hand force sensor and tactile sensor. The paper by C. Wu et al. introduces the application of tactile sensor in prosthetic hand. This paper proposes an EMG prosthetic hand control strategy using force sensor and tactile sensor to improve the control effectiveness and make the prosthetic hand not only controllable but also perceivable. The control strategy consists of EMG self-learning motion recognition, back stepping controller, and force tactile representation. The force and tactile information are not only used for hand grasp control but also for haptic stimulating on user, which helps the user perceive the states of the prosthetic hand.

Robots rely on multiple sensors to provide them with information about their surroundings. Thus, sensor fusion based robot sensing is always a key issue for object tracking, robot path plan and navigation, environment understanding, and autonomous behaviors. The paper by D. Tuvshinjargal et al. proposes a sensor fusion based reactive motion planning method for an autonomous vehicle in dynamic environments. The dynamic motion planning method combines the reactive motion planning technique with a sensor fusion

based obstacle detection approach, which results in improving robustness and autonomy of vehicle navigation within unpredictable dynamic environments. The key feature of the motion planning method is based on a local observer in the virtual plane which allows the effective transformation of complex dynamic planning problems into simple stationary in the virtual plane, and a sensor fusion based obstacle detection algorithm provides the pose estimation of moving obstacles by using a Kinect sensor and a sonar sensor, which helps to improve the accuracy and robustness of the reactive motion planning approach in uncertain dynamic environments.

In the past decades, person tracking system using a robot has achieved a lot of improvements. However, the problems of distinguishing person and reliable following still exist. The paper by S. Jia et al. proposes a person detection and tracking method by representing a person with multicues based on patches and designing a fuzzy based intelligent gear control strategy (FZ-IGS). The person detection algorithm includes a detector and a tracker. The detector divides a person into many patches and represents a patch by the use of multicues including depth, color, and texture. As track evolves, the detector adjusts the person's size according to depth information. By analyzing the depth histograms and patches' similarity with the given person, the detector can easily recognize the occlusion and then make a decision to update the person's appearance model and change the tracking strategy. The tractor based on an extended Kalman filter predicts the person's position as a candidate sample for the detector. Then, the designed FZ-IGS is used to change the turning gain and linear velocity of the robot according to the position of the person from the robot. The FZ-IGS drives the robot towards the person continuously and stably. A conventional method for automated guided vehicle (AGV) localization has certain limitations, such as slip phenomena, because there are variations in the surface of the road and ground friction. Therefore, precise localization is a very important issue for the inevitable slip phenomenon situation. The paper by S.-W. Yoon et al. presents a sensor fusion method to cope with this drawback by using the Kalman filter, which can eliminate the disadvantages of each sensor, such as the image sensor and encoder based sensor, and obtains the precise localization of the AGV in a slip phenomenon situation.

Acknowledgments

At last, we would like to thank all the authors for their valuable contributions and also the reviewers for their critical help, necessary to achieve a high level of papers' quality and thus make the completion of this special issue possible. Also we would like to thank all members of Editorial Board for approving this special issue.

Aiguo Song
Guangming Song
Daniela Constantinescu
Lei Wang
Quanjun Song

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 347379, 7 pages
http://dx.doi.org/10.1155/2015/347379

Research Article

Kalman Filter Sensor Fusion for Mecanum Wheeled Automated Guided Vehicle Localization

Sang Won Yoon, Seong-Bae Park, and Jong Shik Kim

School of Mechanical Engineering, Pusan National University, Jangjeon-dong, Geumjeong-gu, Busan 609-735, Republic of Korea

Correspondence should be addressed to Jong Shik Kim; jskim@pusan.ac.kr

Received 1 December 2014; Accepted 14 January 2015

Academic Editor: Guangming Song

The Mecanum automated guided vehicle (AGV), which can move in any direction by using a special wheel structure with a LIM-wheel and a diagonally positioned roller, holds considerable promise for the field of industrial electronics. A conventional method for Mecanum AGV localization has certain limitations, such as slip phenomena, because there are variations in the surface of the road and ground friction. Therefore, precise localization is a very important issue for the inevitable slip phenomenon situation. So a sensor fusion technique is developed to cope with this drawback by using the Kalman filter. ENCODER and StarGazer were used for sensor fusion. StarGazer is a position sensor for an image recognition device and always generates some errors due to the limitations of the image recognition device. ENCODER has also errors accumulating over time. On the other hand, there are no moving errors. In this study, we developed a Mecanum AGV prototype system and showed by simulation that we can eliminate the disadvantages of each sensor. We obtained the precise localization of the Mecanum AGV in a slip phenomenon situation via sensor fusion using a Kalman filter.

1. Introduction

The Mecanum AGV automated guided vehicle (AGV), which is mounted to a Mecanum wheel that is roller-attached to the axis of rotation with angle of 45°, can move in any direction. Using this special wheel structure, the Mecanum AGV can move in a narrow space and avoid obstacles easily. Thus, it can reduce process time in factory automation. The Mecanum AGV requires an autonomous navigation system to operate in a factory automation environment. In this autonomous navigation system, the core technology is indoor localization. However, a conventional method for a Mecanum AGV has some limits in localization processes, such as the slip phenomenon. Because of the special structure in which the roller is attached to the axis of rotation, the Mecanum AGV frequently slips in the variations of road's surface and ground friction. A conventional method such as the dead reckoning method has accumulated errors because of the inevitable slip phenomenon of the Mecanum wheel [1].

Laser navigation systems have been used in AGV localization sensors. However, the sensor's price is very expensive and the response time is very slow, so a laser system is inappropriate in an indoor navigation system [2]. Other methods for absolute localization include radio frequency identification (RFID) [3], which is an active badge system using infrared light developed at AT&T Labs [4], MIT's cricket system based on ultrasonics [5], and Ubisense Company's Ubitag based on UWB [6]. However, there is no absolute solution regarding localization methods. The sensor fusion method for the mobile robot localization uses a Kalman filter [7, 8] and a particle filter [9, 10]. These methods are based on the Bayesian filter [11]. Many researchers have studied sensor fusion technique using two or more sensors for mobile robot localization; for example, Lee et al. used laser and encoder [12] and Rigatos used sonar and encoder [13]. In this paper, the StarGazer localization sensor (HAGISONIC Co. [14]) was used. As shown in Figure 1, this image sensor analyzes an infrared ray image that is reflected from a passive landmark with an independent ID. This image-based sensor has the advantage of absolute position sensing of the mobile robot. However, moving errors and unexpected errors occur because of landmark misrecognition [15, 16]. On the other hand, the localization method using ENCODER generates accumulated errors, but there are no moving errors

Figure 1: Principle of StarGazer operation.

or unexpected errors. Thus, we obtained the advantages of both types of sensor complementary by using Kalman filter sensor fusion. Also, we prove the precise localization of the Mecanum AGV in inevitable slip phenomenon situation by simulation.

This paper is organized as follows. In Section 2, we describe our analysis of the kinematic modeling of the Mecanum AGV. In Section 3, we describe the Kalman filter sensor fusion algorithms and system modeling. In Section 4, we evaluate sensor fusion algorithms using MATLAB simulation. Finally, our conclusions are given in Section 5.

2. Kinematics Modeling of the Mecanum AGV

A coordinate of a Mecanum AGV with a Mecanum wheel is shown as an "X" shape in the floor plan in Figure 2. Mecanum AGV kinematics were analyzed for each coordinate [15].

We drew the kinematics modeling for Mecanum AGV localization. On the plane, the velocity of the Mecanum AGV $V = [V_X \;\; V_Y \;\; \omega_Z]^T$ can represent the linear velocity of each Mecanum wheel $V_{iW} = [V_{1W} \;\; V_{2W} \;\; V_{3W} \;\; V_{4W}]^T$. That is, the Jacobian equation of the Mecanum AGV kinematics model can represent $V = J^+ V_{iW}$. Also, the reverse inverse Jacobian equation can represent $V_{iW} = JV$.

We obtain (1) by representing the matrix equation as $V_{iW} = JV$:

$$\begin{bmatrix} V_{1W} \\ V_{2W} \\ V_{3W} \\ V_{4W} \end{bmatrix} = J \begin{bmatrix} V_X \\ V_Y \\ \omega_Z \end{bmatrix}, \quad \text{where } J = \begin{bmatrix} -1 & 1 & -(W-L) \\ 1 & 1 & (W-L) \\ 1 & 1 & -(W-L) \\ -1 & 1 & (W-L) \end{bmatrix}. \quad (1)$$

To obtain the inverse of asymmetric matrix J, we used a pseudoinverse matrix $J^+ = (J^T \cdot J)^{-1} J^T$:

$$J^+ = \frac{1}{4} \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{a} & \frac{1}{a} & -\frac{1}{a} & \frac{1}{a} \end{bmatrix}, \quad \text{where } a = (W-L). \quad (2)$$

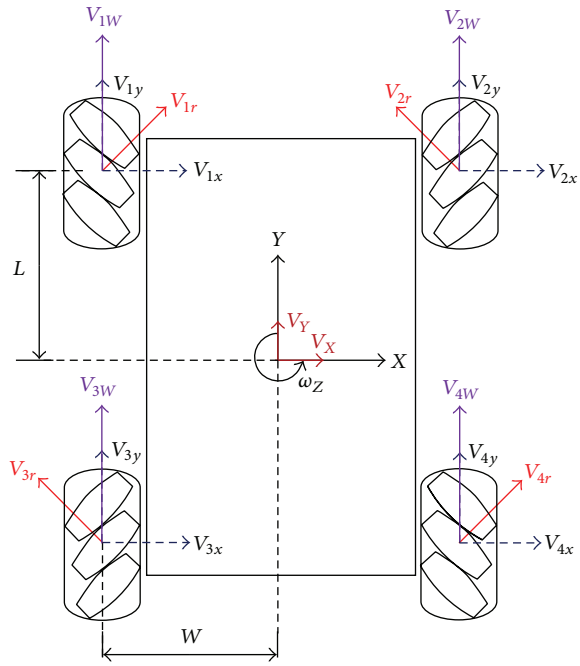


Figure 2: Coordinate of Mecanum AGV.

Therefore, the Jacobian equation of $V = J^+ V_{iW}$ is given by

$$\begin{bmatrix} V_X \\ V_Y \\ \omega_Z \end{bmatrix} = J^+ \begin{bmatrix} V_{1W} \\ V_{2W} \\ V_{3W} \\ V_{4W} \end{bmatrix}, \quad \text{where } J^+ = \frac{1}{4} \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{a} & \frac{1}{a} & -\frac{1}{a} & \frac{1}{a} \end{bmatrix}. \quad (3)$$

Each wheel's linear velocity is the product of the angular velocity and the radius of the wheel. Thus, $V_{iW} = R\dot{\theta}_i$, where R is Mecanum wheel radius and $\dot{\theta}_i$ is the wheel's angular velocity. This equation can be represented as

$$\begin{bmatrix} V_X \\ V_Y \\ \omega_Z \end{bmatrix} = \frac{R}{4} \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{a} & \frac{1}{a} & -\frac{1}{a} & \frac{1}{a} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix}. \quad (4)$$

Now, $[V_X \;\; V_Y \;\; \omega_Z]^T = [\dot{X}_X \;\; \dot{Y}_Y \;\; \dot{\theta}_Z]^T$ is a moving coordinate so we must translate the reference coordinate using a transformation matrix:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_X \\ V_Y \\ \omega_Z \end{bmatrix}. \quad (5)$$

We can obtain the position by integrating this velocity of the reference coordinate $[\dot{x}_r \;\; \dot{y}_r \;\; \dot{\theta}]^T$. This method is called dead

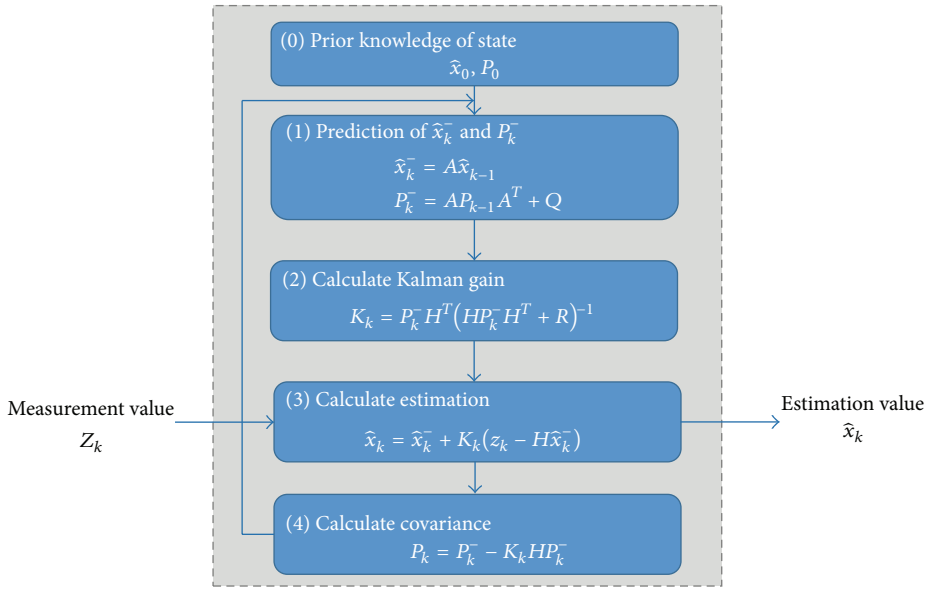


Figure 3: Kalman filter algorithm.

reckoning, and a conventional mobile robot position system typically uses this method.

3. Kalman Filter Sensor Fusion Algorithms for Localization

The dead reckoning method described in Section 2 has inevitable accumulated errors because of the Mecanum wheel's mechanical structure and variations in the road's surface. So, this approach can be used for short distances but it cannot be used for long distances and path following. Generally, long distances and path following use a sensor fusion technique using dead reckoning and another sensor for localization [16].

A Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The filter is named after Rudolf (Rudy) E. Kálmán, one of the primary developers of its theory [17]. The Kalman filter has numerous applications in technology. Common applications include guidance and the navigation and control of vehicles, particularly aircraft and spacecraft.

The Kalman filter algorithm consists of four processes, as shown in Figure 3. The algorithm includes prediction and estimation functions.

(1) Estimation. This is the first step, shown in Figure 3. Input value using prior estimation value ($\hat{x}_{k-1}$) and covariance (P_{k-1}^-), and finally calculate estimation value ($\hat{x}_k^-, P_k^-$). These values will be used in the prediction step.

(2) Prediction. This involves the second, third, and fourth steps shown in Figure 3. The final value of these steps is an estimation value ($\hat{x}_k$) and the covariance (P_k). Input values use estimation value ($\hat{x}_k^-, P_k^-$) of estimation step result and measurement value (z_k).

Here covariance (P_k) is criterion, that is, difference between real value and estimation value of Kalman filter:

$$x_k \sim N(\hat{x}_k, P_k). \tag{6}$$

This means variables x_k mean normal distribution that average value is $\hat{x}_k$ and covariance is P_k. Kalman filter algorithms choose estimation value using probability distribution of estimation value x_k which becomes the maximum probability value.

Figure 4 shows a Kalman filter sensor fusion-based encoder and StarGazer for Mecanum AGV localization in which the system model is ENCODER and the observation model is StarGazer.

The Kalman filter discrete system modeling and observation modeling based on Jacobian equation (3) of the kinematics modeling of the Mecanum AGV are described as follows.

(i) ENCODER-Based System Modeling. Consider

$$\begin{bmatrix} V_X & V_Y & \omega_Z \end{bmatrix}^T = \begin{bmatrix} \dot{X}_X & \dot{Y}_Y & \dot{\theta}_Z \end{bmatrix}^T, \tag{7}$$

$$\begin{bmatrix} X_X \\ Y_Y \\ \theta_Z \end{bmatrix}_k = \begin{bmatrix} X_X \\ Y_Y \\ \theta_Z \end{bmatrix}_{k-1} + \frac{R}{4} \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{a} & \frac{1}{a} & -\frac{1}{a} & \frac{1}{a} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix}_k \tag{8}$$

$$\times T + w_k, \quad \text{where } T\text{: Sampling Time,}$$

$$x_k = Ax_{k-1} + Bu_k, \quad \text{where } A = I_3,\ B = \frac{R}{4}J^+,$$

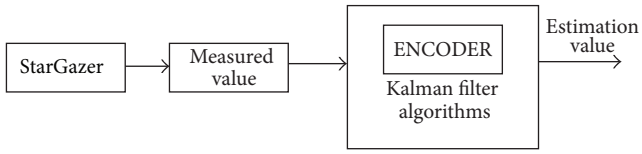


Figure 4: Diagram for Kalman filter sensor fusion.

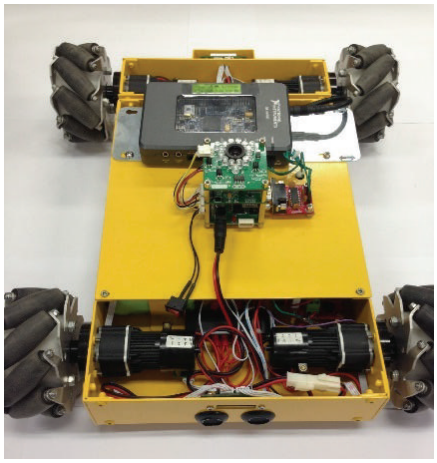


Figure 5: Prototype of Mecanum AGV.

where $\dot{\theta}_i$ is control input u_k, which is the angular velocity measured by ENCODER. The $[X_X \;\; Y \;\; \theta_Z]^T$ is the moving coordinate. So, using the transformation matrix, a transform is made to the reference coordinate system:

$$\begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_X \\ Y_Y \\ \theta_Z \end{bmatrix}. \tag{9}$$

(ii) StarGazer-Based Observation Modeling. The observation model is the StarGazer sensor-based position value x_k $(X_X \;\; Y_Y \;\; \theta_Z)$. Observation model is designed by including disturbance (v_k):

$$z_k = Hx_k + v_k, \quad \text{where } H = I_3,\ v_k = \text{Experiment Value}, \tag{10}$$

where the experiment value v_k is the measurement noise of StarGazer following $v_k \sim N(0, 0.1^2)$. Following $v_k \sim N(0, 0.1^2)$ is a normal distribution value where the mean value is 0 and the standard deviation is 0.1.

The system model and observation model use Kalman filter sensor fusion, where the position acquired by StarGazer is in the measured value and (8) represents the system model. Therefore, the Kalman filter sensor fusion attempts to eliminate the StarGazer position error-based ENCODER system model using Kalman filter algorithms.

Table 1: Specification of Mecanum AGV.

Chassis	Length	400 mm
	Width	360 mm
	Wheel Base	300 mm
	Speed	0.6 m/s
Wheel	Type	Mecanum wheel
	Diameter	100 mm
Motor	Type	12 V DC coreless motor
	RPM	120 rpm

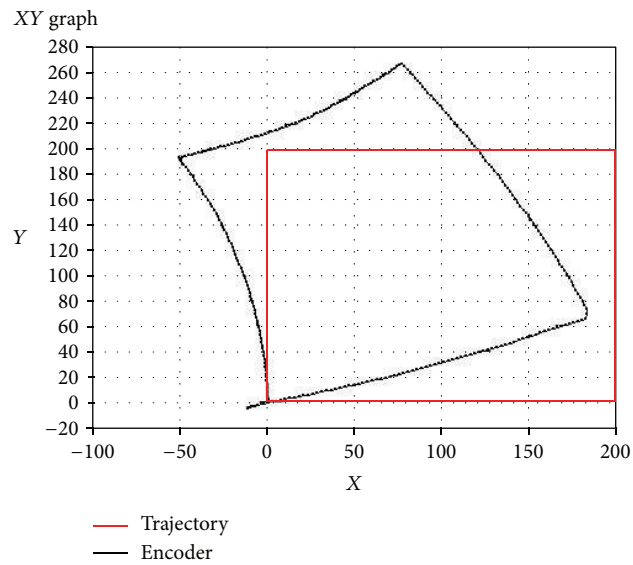


Figure 6: Mecanum AGV localization for ENCODER.

4. Experiment and Simulation Results

4.1. Mecanum AGV Localization System Prototype. Figure 5 shows our prototype of the Mecanum AGV for localization. MyRIO (NI Co.) was used for sensor data acquisition. We developed a code to allow ENCODER and StarGazer sensor data to use LabVIEW.

Specifications for the chassis, wheels, and motor are shown in Table 1. The Mecanum wheel and roller were made of aluminum and synthetic rubber, and the gear ratio is 64 : 1. Two MAI-2MT-DC drivers were used as drive motors. The optical encoder has 12 CPR (count per revolution) resolution. StarGazer obtains position data ten times per second by RS232 using MyRIO.

4.2. Localization by ENCODER. The localization experiment using ENCODER was composed that Mecanum AGV drives 2 m width and height square path and then integrates the velocity of the reference coordinate $[\dot{x}_r \;\; \dot{y}_r \;\; \dot{\theta}]^T$ o of (5). Figure 6 shows that the localization by ENCODER has accumulation errors because of integration errors and slip phenomenon. Nevertheless, ENCODER has the advantage that there is no dead-zone, and there are no moving errors.

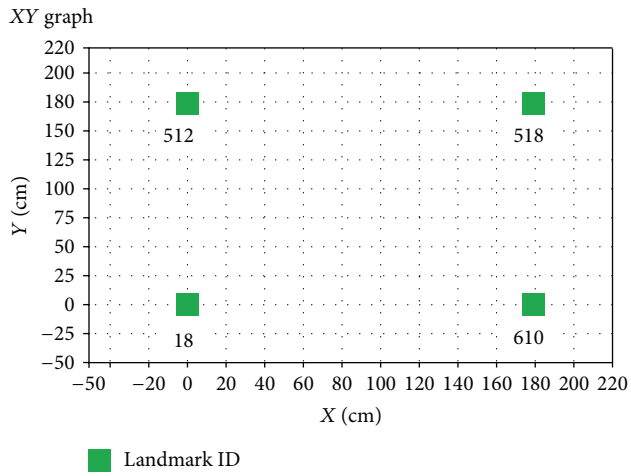


Figure 7: Landmark ID of StarGazer.

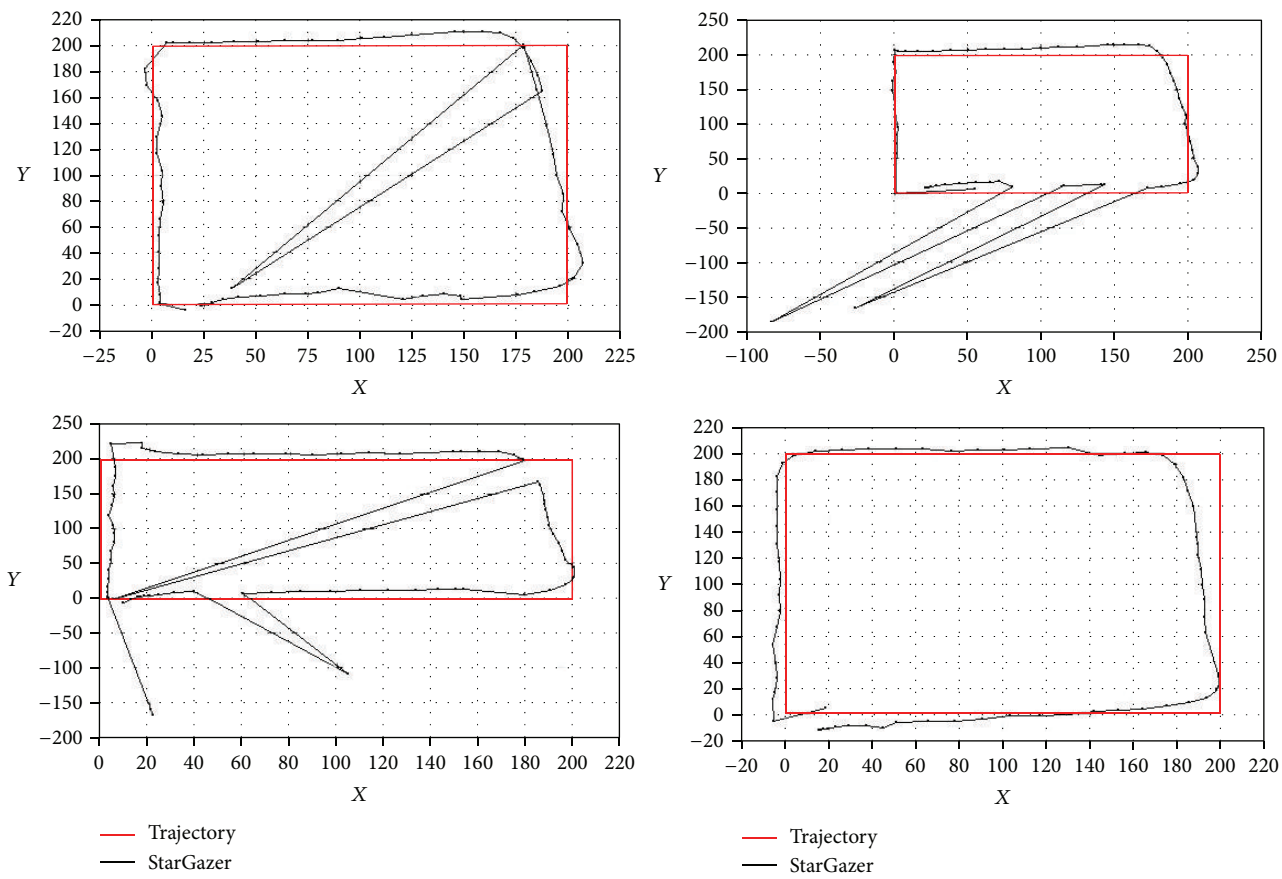


Figure 8: Mecanum AGV localization for StarGazer.

4.3. Localization by StarGazer. In this localization experiment, four landmark IDs are arranged in a square layout as shown in Figure 7. Each landmark space out 1.8 m, and each landmark can measure data within a 2 m radius. StarGazer detected one Landmark with respect to another Landmark. Then, the position value was calculated for the relative coordinate of each landmark.

Figure 8 shows position data calculated by StarGazer. The Mecanum AGV drove 2 m space square path. A total of 30 experiments were performed, and four experiment cases are shown in Figure 8. The average moving error is less than about 10 cm. As shown in Figure 8, unexpected large position errors occurred. This is caused by StarGazer that can be confused in overlapped area between one landmark and

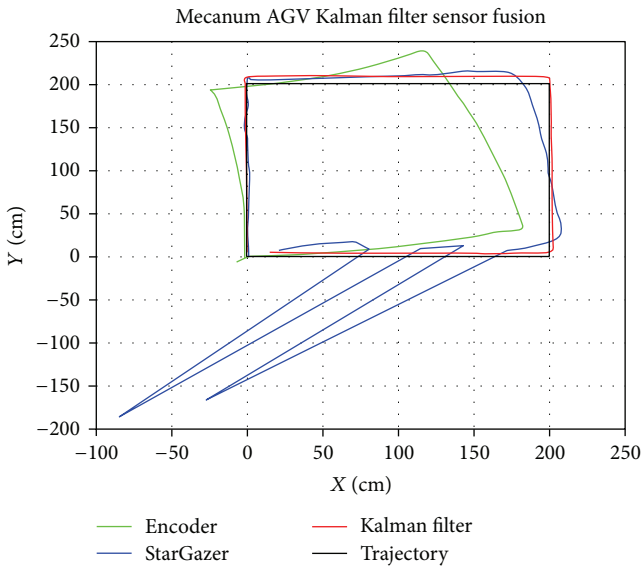


Figure 9: Square path experiment results of Mecanum AGV.

another. In experiment average unexpected large position errors occurred 1.4 times of 30. These large position errors caused unscented moving of the AGV. Because of these errors, misunderstanding positions of AGV causes wrong path following. So, these terms must be deleted in order to follow the correct path. StarGazer, in comparison to ENCODER, can calculate the absolute value of the mobile robot with accumulated errors.

4.4. Kalman Filter Sensor Fusion Simulation Result. In this experiment, the Mecanum AGV drove a square path (width 2 m, height 2 m) in an area with dimensions of 4 m × 4 m. The measured values acquired by ENCODER and StarGazer were used in simulation by MATLAB. Then, the localization value from ENCODER by integrating (7), the localization value from StarGazer, and the Kalman filter sensor fusion value were compared.

As shown in Figure 9, the green line represents the localization value by ENCODER and the blue line represents the localization value by StarGazer. The red line represents the Kalman filter sensor fusion localization value. Sensor fusion can delete accumulated error of ENCODER and the large position error because of the landmark misrecognition problem. These results mean that localization errors by StarGazer were deleted from using Kalman filter system modeling based on Mecanum AGV kinematics model (9). The resulting estimation path from the Kalman filter sensor fusion deviated from the 2 m square path due to the Mecanum AGV slip phenomenon.

5. Conclusion

This paper was written for sensing precise localization values of Mecanum AGV nevertheless unavoidable slip phenomenon. To overcome this phenomenon, we used two sensors: StarGazer and ENCODER. StarGazer can measure absolute localization values but produces large errors because of landmark misrecognition. ENCODER does not have moving errors and there is no dead zone, but there are accumulated errors because of the integrating term and slip phenomenon. A Kalman filter was also used to obtain the advantages of both types of sensors. Our simulation results show that the Kalman filter sensor fusion method can delete accumulated errors of ENCODER and StarGazer moving error and big error caused landmark misrecognition. Mecanum AGV for autonomous driving can move narrow path and sideway moving can easily approach conveyer line. For autonomous driving, core technology is localization. This method can suggest the Mecanum AGV localization solution to overcome such as unavoidable slip phenomenon by sensor fusion ENCODER and StarGazer. The proposed method in this paper is expected to bring innovation to factory automation. Future research topics include path tracking, path following, and the map building process method using Mecanum AGV.

Conflict of Interests

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

References

[1] J.-M. Kim, S.-B. Woo, J.-Y. Kim, J.-C. Do, S.-S. Kim, and S.-I. Bae, "Inertial navigation system for an automatic guided vehicle with Mecanum wheels," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 3, pp. 379–386, 2012.

[2] J. Park, *Localization based sensor fusion for driving control of robot with mecanum wheel [M.S. thesis]*, Pusan National University, Busan, Republic of Korea, 2011.

[3] L. Jing and P. Yang, "A localization algorithm for mobile robots in RFID system," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 2109–2112, September 2007.

[4] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.

[5] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 32–43, August 2000.

[6] http://www.ubisense.net.

[7] S.-P. Kim, *Essential Kalman Filter*, A-JIN, Seoul, Republic of Korea, 3rd edition, 2014.

[8] R. Negenborn, *Robot localization and kalman filters: on finding your position in a noisy world [M.S. thesis]*, Utrecht University, Utrecht, The Netherlands, 2003.

[9] F. Gustafsson, F. Gunnarsson, N. Bergman et al., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[10] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, New York, NY, USA, 2001.

[11] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Bordello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, 2003.

[12] I. S. Lee, J. Y. Kim, J. H. Lee, and S. Kim, "Kalman filter-based sensor fusion for improving localization of AGV," *Advanced Materials Research*, vol. 488-489, pp. 1818–1822, 2012.

[13] G. G. Rigatos, "Extended Kalman and particle filtering for sensor fusion in motion control of mobile robots," *Mathematics and Computers in Simulation*, vol. 81, no. 3, pp. 590–607, 2010.

[14] http://www.hagisonic.com.

[15] M.-S. Chae and T.-G. Yang, "A study on precise localization for mobile robot based on artificial landmarks in the ceiling," *Journal of Advanced Information Technology and Convergence*, vol. 9, no. 8, pp. 85–92, 2011.

[16] S.-J. Park and T.-g. Yang, "A study on design of intelligent mobile robot based on localization sensor for unmanned transport logistics," *Journal of KIIT*, vol. 11, no. 9, pp. 7–13, 2013.

[17] S. G. Tzafestas, *Introduction to Mobile Robot Control*, Elsevier, London, UK, 1st edition, 2014.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 760435, 12 pages
http://dx.doi.org/10.1155/2015/760435

Research Article

Person Tracking System by Fusing Multicues Based on Patches

Song Min Jia,[1,2,3] Li Jia Wang,[1,2,4] Xiu Zhi Li,[1] and Lin Feng Wen[1]

[1] *College of Electronic Information & Control Engineering, Beijing University of Technology, No. 100, Pingleyuan, Chaoyang District, Beijing 100124, China*
[2] *Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China*
[3] *Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China*
[4] *Department of Information Engineering and Automation, Hebei College of Industry and Technology, No. 626, Hongqi Street, Qiaoxi District, Shijiazhuang 050081, China*

Correspondence should be addressed to Song Min Jia; jsm@bjut.edu.cn

Received 15 January 2015; Revised 13 May 2015; Accepted 20 May 2015

Academic Editor: Yasuko Y. Maruo

A person tracking algorithm by fusing multicues based on patches is proposed to solve the problem of distinguishing person, occlusion, and illumination variations. Kinect is mounted on the robot for providing color images and depth maps. A detector representing a person by using the fusion of multicues based on patches is proposed. The detector divides the person into many patches and then represents each patch by using depth-color histograms and depth-texture histograms. The appearance representation, considering depth, color, and texture information, has powerful discrimination ability to handle the problems of occlusion, illumination changes, and pose variations. Considering the motion of the robot and person, a tracker called motion extended Kalman filter (MEKF) is presented to predict the person's position. The result of the tracker is treated as a candidate sample of the detector, and then the result of the detector is the previous knowledge of the tracker. The detector and tracker supplement each other and improve the tracking performance. To drive the robot towards the given person precisely, a fuzzy based intelligent gear control strategy (FZ-IGS) is implemented. Experiments demonstrate that the proposed approach can track a person in a complex environment and have an optimum performance.

1. Introduction

With the popularity of robot in human environments, it is necessary to detect and track a person in many applications including surveillance, search, rescue, combat, and human assistant. Person detecting and tracking are very challenging computer vision tasks due to automatic initialization, pose variations, expensive calculation cost, and occlusions in complicated environments [1].

In real-world settings, persons are nonrigid and difficult to be tracked. To resolve the problem, an efficient representation should be considered for an available appearance model. Color is widely used for modeling a target, and one of the best methods for color-based object tracking is to realize the mean shift algorithm [2, 3]. Ning et al. [4] presented a scale and orientation adaptive mean shift algorithm to handle the problem of scale and orientation changes. Unfortunately, the pixel-wise color density does not consider extreme geometric changes of an object. It is vulnerable when there is occlusion or similar background. Many researches have focused on resolving the problem by utilizing the texture feature. Ning et al. [5] used joint color-texture histograms for robustly tracking a target in complex environment. Compared with the traditional color histogram, the joint color-texture histograms efficiently exploited a target's structure information and hence performed better when a target has similar color appearance with the background.

To further eliminate the influence of background, depth information captured from stereo cameras is employed. The depth information easily performs the foreground-background segmentation [6]. However, most stereo tracking systems are implemented with known calibration parameters [7]. In the last few years, stereocameras (e.g., Kinect [8]), with no extensive knowledge of camera calibration parameters

and low cost, have been widely used in computer vision [9, 10]. Compared with the traditional stereocameras [7], Kinect can provide higher quality color image and depth map and is widely employed recently. Xia et al. [10] considered object tracking as foreground-background segmentation by extracting contour information and depth feature from a Kinect sensor. Zoidi et al. [6] represented an appearance by fusing Local Steering Kernel features and 2D color-disparity histograms. The method employed disparity information to identify scale changes by analyzing disparity values. The depth image (disparity image) indicates objects' distances in the complexity environment, which meets the human visual perception system. Therefore, the depth information is of great significance to discriminate the target from the background.

Occlusion is a difficult problem in object tracking. To cope with the problem, patches based algorithms were proposed [11–14]. Adam et al. [11] presented a fragments-based color histograms method. The method represented a target by integrating each part's color histograms to handle partial occlusions and pose variations. Nejhum et al. [12] used multiple blocks to model a frequently changing foreground shape. The method successfully tracked objects undergoing significant shape variations and illumination changes. Yang et al. [13] proposed a spatially attentional patches based tracking method which performed well on a large number of real-world videos. Kwon and Lee [14] proposed a patches based dynamic appearance model for representing a target. The hue, saturation, and value features were adaptively selected for calculating photometric likelihood, while the squared differences between patches were adopted for representing geometric likelihood. Unfortunately, it suffered from a high computing burden due to the Basin Hopping Monte Carlo.

For a robot system in clustered environment, a continuous and stable controller is important for following a person. However, to the author's knowledge, many works have focused on the problem of target detection and tracking but rarely addressed the problem of designing a suitable controller for driving a robot [15]. The existing controller mainly includes the PID controller [16], visual based sliding mode controller [15], fuzzy based controller [17], and intelligent controller [7]. Ouadah et al. [15] presented two sliding mode controllers to control the robot according to the person's position obtained from RFID system and visual system, respectively. The robot can follow the given person when there is a sudden turn. Jia et al. [7] presented an intelligent speed controller considering the robot's kinematics. However, the algorithm often fails to follow the person because of the fixed linear velocity.

In the past decades, person tracking system using a robot has achieved a lot of improvements. However, the problems of distinguishing person, occlusion, and safe following still exist. We address the problem by representing a person with multicues based on patches and designing a fuzzy based intelligent gear control strategy (FZ-IGS). The person detection algorithm includes a detector and a tracker. The detector divides a person into many patches and represents a patch by the use of multicues including depth, color, and texture. The depth information, indicating the person's location, is combined with color and texture features for generating depth-color histograms and depth-texture histograms, respectively. As track evolves, the detector adjusts the person's size according to depth information. By analyzing the depth histograms and patches' similarity with the given person, the detector can easily recognize the occlusion and then make a decision to update the person's appearance model and change the tracking strategy. When there is a partial occlusion, the detector recognizes the person by using the patches which are not occluded. The tractor called MEKF is generated from the EKF by considering the motion of the robot and person. The MEKF predicts the person's position as a candidate sample for the detector. Finally, FZ-IGS is designed to change the turning gain and linear velocity of the robot according to the position of the person from the robot. The FZ-IGS drives the robot towards the person continuously and stably.

The paper is organized as follows: the overview of the proposed method is discussed in Section 2. In Section 3, the multicues based detector is described. Section 4 details the steps of processing person location and model update. The fuzzy based controllers are described in Section 5. The experimental results are detailed in Section 6. The paper conclusion with a short summary is shown in Section 7.

2. Framework and Architecture

The section details the platform and the system overviews for performing the person following task.

2.1. Development Platform and Environment. The platform used for performing person following task is an American Mobile Robots Inc. Pioneer 3-DX embedded with a Kinect, illustrated in Figure 1. The Kinect is a new and widely available device for the Xbox 360. The interest for Kinect is increasing in computer vision due to its advantages of providing 3D information of the environment and low cost. The device contains an RGB camera, a multiarray microphone, and a depth sensor. Using these sensors, Kinect can capture full body 3D motion. The Kinect hardware specification is detailed as follows:

(1) RGB camera: 640 × 480 pixels/32 bit colour at 30 frames/sec,

(2) depth sensor: 320 × 240 pixels/16 bit greyscale at 30 frames/sec,

(3) sensor range: 1.2 m–3.5 m,

(4) field of view: horizontal: 57° (1.3 m–3.8 m); vertical: 43°.

Using these sensors, the Kinect can provide two kinds of images: depth image and color image. The depth image is obtained by the depth sensor which contains a CMOS camera and an infrared projector. The infrared projector produces speckle pattern in the scene. Then, the CMOS camera records the speckle pattern and results in the depth image. The color image is produced by the RGB camera with a resolution of 640 × 480 pixels at 30 frames per second. The Kinect has a

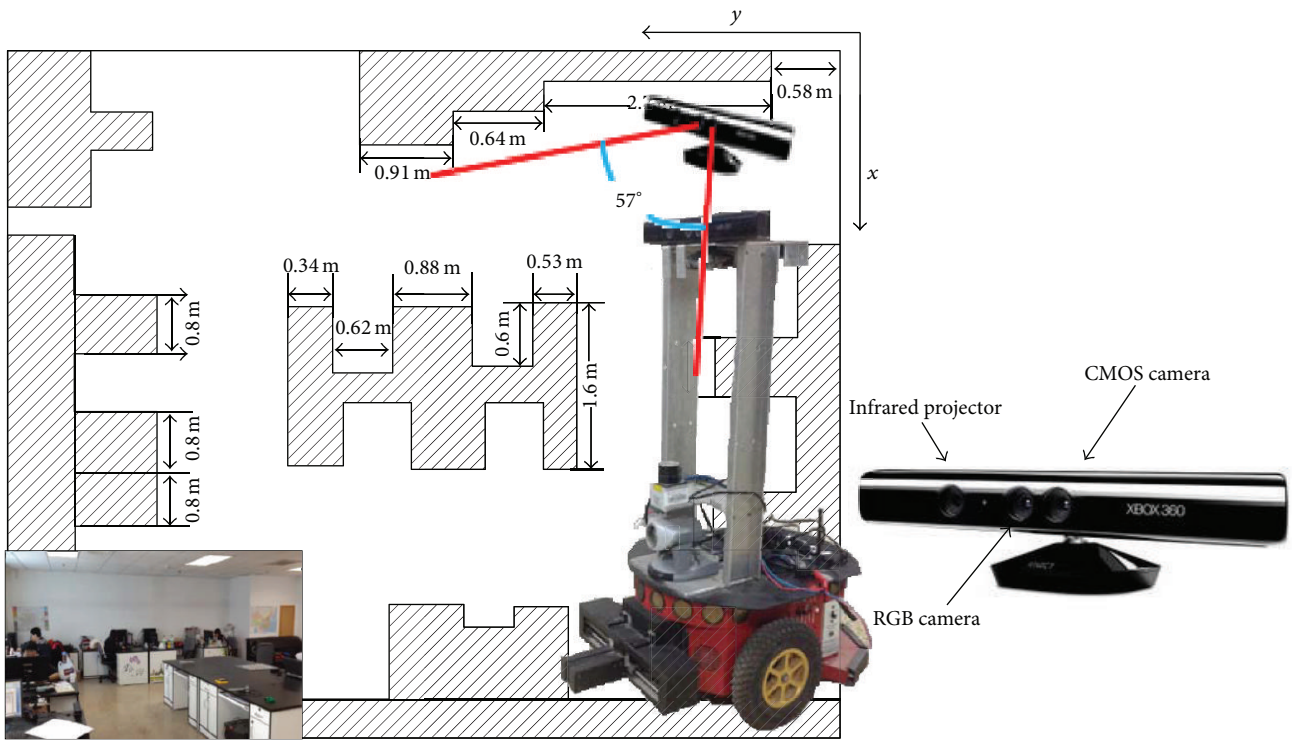


Figure 1: The platform for person tracking.

field of view 57° which can satisfy the need of object tracking. The algorithm is implemented by VC++2008 and Opencv2.1.

2.2. System Overview. Given a stream of color images and depth maps, our goal is to continuously track a person. The overview of our system is presented in Figure 2. The system includes a detector, a tracker, and an online update strategy. The detector represents the person by using the depth, color, and texture information obtained from the Kinect. The tracker predicts the person's position by considering the person and robot's motion. The result of the tracker is treated as a candidate sample of the detector for determining the person's location, and then the result of the detector is adopted as the previous information of the tracker. The detector and tracker supplement and complement each other, which improves the tracking performance. As track evolves, the detector adjusts the person's size according to depth histograms and determines the occlusion problem based on the depth histograms and patches' appearance similarity. Finally, the online update strategy adaptively updates the person's appearance to avoid introducing more inference and handle the variations on illumination and pose.

3. Detector

It is reported that the appearance represented by a single feature often fails in tracking process when there is similar background. To handle the problem, we represent a person by using multicues including depth, color, and texture. The detector can successfully recognize a person by using one feature while the other features are invalid. The depth feature, easily discriminating the person from background, is extracted for representing the person to overcome the background's inference. Furthermore, the detector detects the problem of occlusion considering the depth histograms and the patches' appearance similarity and then adjusts the online update strategy.

3.1. Depth Histograms. Depth map, captured from the sensor Kinect, provides 3D information of the environment and is invariant to illumination [6]. Compared with the color image, the depth values provide an intuitive notion of the relative person's distance from the robot. The larger the depth value is, the closer the person is to the robot. In our case, the robot is controlled towards the given person and remains in a safety distance from him. Therefore, the depth values of the person are closest to the robot. Then, the depth segmentation can be performed on the depth map for distinguishing the person from the background.

The depth is discretely distributed in n intervals. The depth values are represented by a vector $x_{i\ i=1,2,\ldots,M}^{*}$; M is the number of the depth value. A delta function $\delta(\cdot)$ is employed to determine the interval for the depth value x_i. Then, depth features, called depth histograms, are extracted by analyzing the pixels of the depth image:

$$\widehat{q}_d = \sum_{i=1}^{M} \delta\left[b\left(x_i^*\right) - u\right], \tag{1}$$

where u is an interval. During tracking, we assume that the person is in front of the robot and his position from the robot

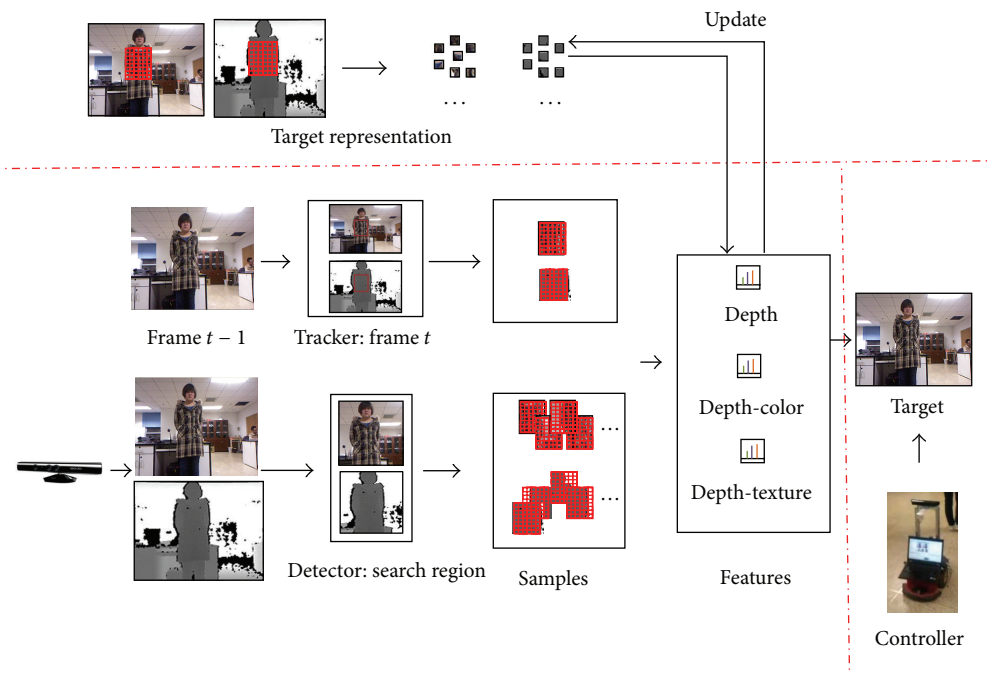


Figure 2: The overview of the system.

does not change significantly from frame to frame. Therefore, all of the person's depth values will lie in the last bin of the depth histograms and will be far from the background. Considering the depth histograms, the foreground-background segmentation will be much easier. The depth histogram is shown in Figure 3. Affected by the illumination, the depth value in some region is higher than another region, shown in Figure 3(a). Therefore, the bins belonging to the target are the last two bins (181 and 211) shown in Figure 3(b). The depth histogram for the target in the blue rectangle is shown in Figure 3(c). It has two bins: bin 181 is for the region with lower depth value; bin 211 is for the region with higher depth value.

Furthermore, the person's size changes according to the variation of his position from the robot in the tracking process. The appearance model obtained by using the fixed rectangle size will introduce background's inference or lose some important information when the distance changes. While the distance is large, we expect the rectangle size to be small for fitting the person. When the person is close to the robot, we expect the rectangle size to be large to fit the person. The depth information indicates the changes of the person's size. In the case in which the person's position changes, the bin values of the depth histograms will correspondingly vary. Thus, we adaptively adjust the rectangle's size based on the depth histograms. The person's current size is obtained as follows:

$$\text{size}_{\text{new}} = \text{size}_{\text{old}} \times \gamma, \tag{2}$$

where $\gamma = \hat{q}_d / \hat{q}_{\text{base}}$ and $\hat{q}_d$ is the target's depth histogram. $\hat{q}_{\text{base}}$ is the reference depth feature which is determined by the initial object. c_{base} is the adjustment parameter which is determined by initial size of the target. $\text{size}_{\text{old}} = \{W, H\}$ is the size of the person in previous frame. When the person's size changes due to the variations of the distance between the person and the robot, the rectangle size is updated. The obtained person's rectangle size, fitting the variations of the distance, can not only avoid inducing more inference from background due to a larger rectangle but also avoid losing important information because of the smaller rectangle. After adjusting the person size, the detector collects the candidate samples based on the new size and updates the appearance model.

3.2. Depth-Color/Texture Detector. In order to successfully discriminate a given person, multicues are employed for representing the person. Color has been proved to be useful for modeling a target. Compared with other features, color is insensitive to scale and translation. Therefore, it has been widely adopted for target representation. Texture, as another effective description operator, indicates the pixels' space property. To obtain more powerful representation, color and texture are mixed for modeling a target.

The traditional color and texture based object representation has successfully discriminated person when there is color or texture clutter in background [5]. However, it can hardly solve the problem of occlusion and complex background. In our research, depth information is employed to handle color or texture clutter in background due to the depth's ability of foreground-background segmentation. The disadvantage of depth segmentation is that it cannot easily discriminate the objects lying in the same distance from the robot. Fortunately, this can be resolved by using the color or texture features. The depth, color, and texture are combined to generate the depth-color histograms and depth-texture histograms for representing the person. Moreover, to deal with the occlusion problem, patches based representation method is presented.

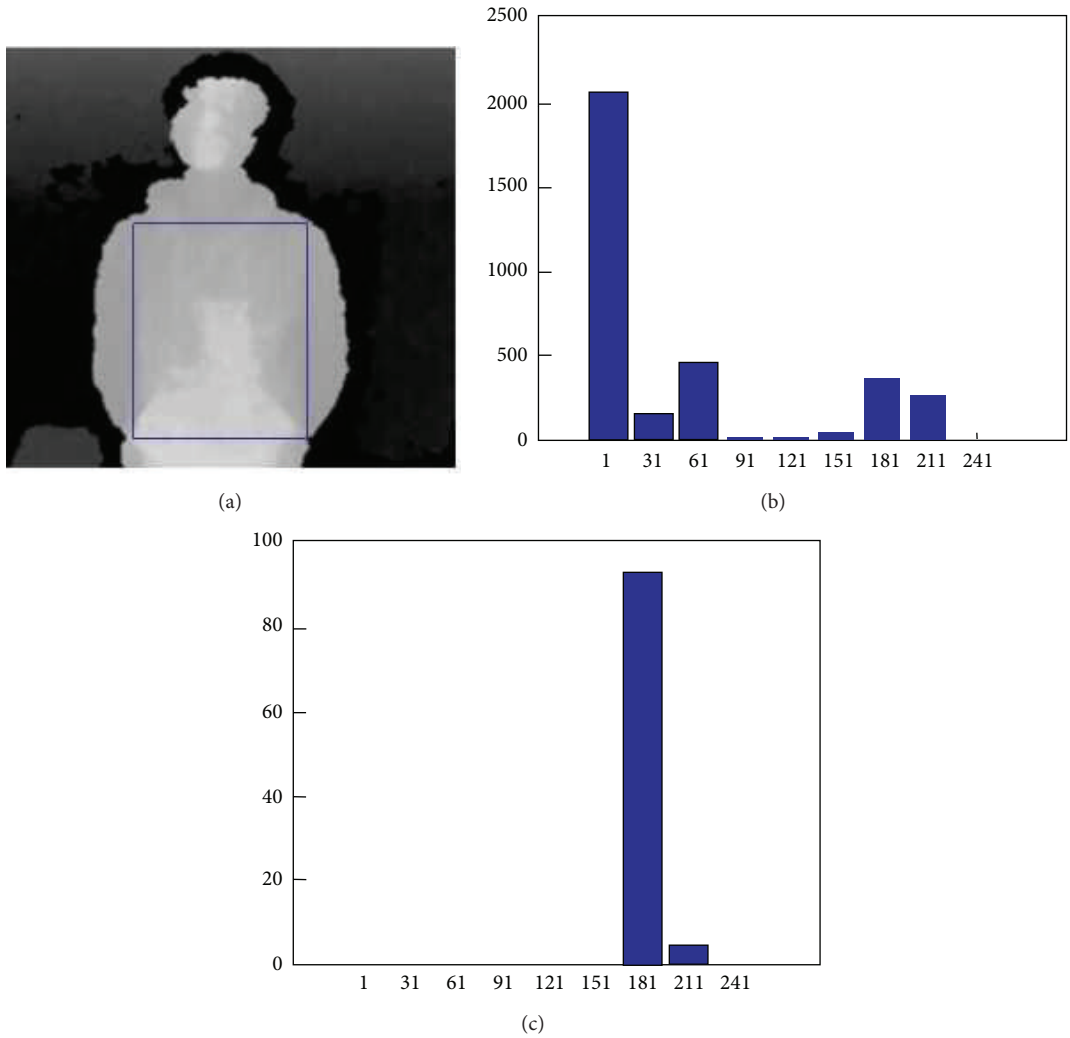


Figure 3: The illustration of the depth histogram. (a) The depth image, the blue rectangle is for the target. (b) The depth histogram for the depth image. (c) The depth histogram for the target in the blue rectangle.

The person in a rectangle is divided into $N \times N$ patches and each patch is represented by the depth-color and depth-texture histograms:

$$\hat{q}_{f,n,u} = C\sum_{i=1}^{M} K\left(y_0, x_i^*\right)\delta\left[b\left(x_i^*\right) - u\right], \tag{3}$$

where $f = Cd, Ld$ indicates the depth-color and depth-texture information, respectively. $\hat{q}_f$ is the obtained depth-color histograms and depth-texture histograms. The color feature is captured from the HSV space, while the texture information is the uniform texture [5]. $n = N \times N$ is the number of the person's patches. $C = 1/\sum_{i=1}^{M} K(y_0, x_i^*)$ is the normalized coefficient. $\{x_i^*\}_{\{i=1,\ldots,M\}}$ is the pixels of each patch; y_0 is the center of each patch. $\delta(\cdot)$ is the *delta* function for determining the feature's bin number. $K(y_0, x_i^*)$ is a kernel function which affects the obtained features' discriminative power. The Epanechnikov function is commonly used. It assigns a larger weight for the pixels in the center of the target image and a smaller value for the pixels far away from the center. This method can avoid introducing to a certain extent the inference of the background around the person. However, for the pixels far away from the center, its importance in the appearance representation is reduced due to the smaller weight. Furthermore, the edges and background far away from the center of an irregular target (e.g., person) may be confused. In such a case, the pixels in the background with smaller weights are introduced into the appearance model. To deal with the problem, depth information is used for constructing the new kernel function for segmenting the target from the background:

$$K\left(y_0, x_i^*\right) = M_{\text{depth},n}\left(x_i^*\right), \tag{4}$$

where

$$M_{\text{depth},n}\left(x_i^*\right) = \begin{cases} g\left(u^*\right), & x_i^* \in \hat{q}_d\left(u^*\right), \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

is the mask image. u^* is the bin belonging to the person. The new kernel function avoids introducing the background's inference because the pixels with the value $g(u^*)$ in the mask

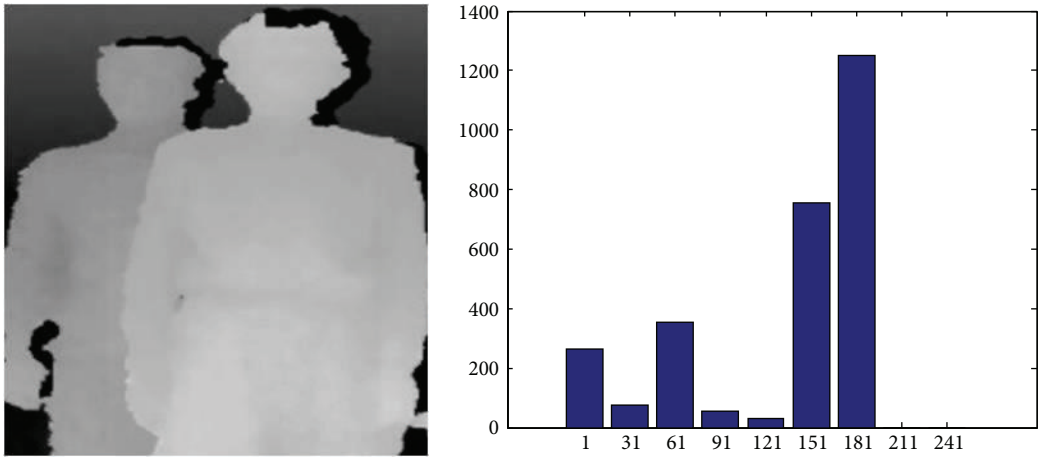


Figure 4: The illustration of depth information when there is occlusion.

image belong to the person and the pixels with the value 0 belong to the background. Compared to the Epanechnikov function, the new function assigns the pixels of the person the same weights to improve the representation's discrimination ability.

The person is represented by modeling each patch using the obtained depth-color and depth-texture histograms. The color histograms describe the target integrally, while the texture histograms depict image's local texture. The two features somehow supplement each other. The depth information, identifying the target from background, deals with the color or texture clutter in background.

3.3. Occlusion Problem. As tracking evolves, there may be occlusion which will result in tracking failure. The patches based tracking algorithm was proposed to deal with the problem [18]. The person is divided into $N \times N$ patches. When there is partial occlusion, some patches are occluded and others are free. We present a method to detect the occlusion problem by using the patches based appearance similarity and the depth histograms. After detecting the occlusion, the person is discriminated by processing the unoccluded patches. The depth map with occlusion problem is shown in Figure 4.

For a person tracking system, the person's depth information lies in the last bin of the depth histograms and is far away from the background usually. In the case in which the person is occluded, the last two bins are next to each other. The last bin belongs to the passerby, while the last bin but one is for the person. As shown in Figure 4, the last bin "181" with more than 1200 depth values belongs to the passerby that is near the camera. The last bin but one "151" is for the person that is occluded by the passerby. Then, for the person and the passerby, we perform depth feature and the patches' similarity for detecting the occlusion problem.

In an ideal tracking process, the bin for the person maintains stability. The depth feature similarity is calculated as $S_q = \widehat{q}_d - \widehat{q}_d^t$, where $\widehat{q}_d^t$ is the depth feature in the current frame. A threshold is set to determine whether the depth feature belongs to the target. Moreover, the patches' similarity is processed to detect the person successfully, which will be shown in Section 3.4.

3.4. Patches Based Multicues for Person Detection. Compared with only one feature, to represent a person by extracting different features can improve the model's discrimination ability. Once one feature fails to discriminate the person, the other features are valid. The person is represented by many patches which are in a decreasing order based on the depth-color histograms and depth-texture histograms, respectively. For a given threshold, th, the detector recognizes the person according to their appearance similarity. Normally, the candidate sample with the maximum overall similarity and over 90% of the number of patches is the person. However, some features (e.g., depth-color histograms) may change much due to pose variations or illumination changes. Then, the corresponding similarity will decrease and the overall similarity will be less than the threshold, th. In such a case, the detector will recognize the person based on the other feature's similarity (e.g., depth-texture histograms) and the number of the patches. Once partial occlusion is detected, the detector recognizes the person according to the patches which are not occluded. The patches based multicues representation is shown in Figure 5.

The patch's similarity between the candidate sample and the person is measured by using the cosine similarity metric:

$$\widehat{\rho}_{f,n}\left(\widehat{p}_{f,n}, \widehat{q}_{f,n}\right) = \cos(\theta) = \frac{\left\langle \widehat{p}_{f,n}, \widehat{q}_{f,n} \right\rangle}{\left\| \widehat{p}_{f,n} \right\| \left\| \widehat{q}_{f,n} \right\|} \in [-1, 1], \quad (6)$$

where $\langle \cdot \rangle$ is the inner product and $\| \cdot \|$ indicates the Euler distance. θ is the angle between two vectors.

The similarity between the candidate sample and the model is

$$\widehat{\rho}_f = \sum_{n=1}^{N \times N} \frac{\widehat{\rho}_{f,n}^2}{1 - \widehat{\rho}_{f,n}^2} \in [0, +\infty]. \quad (7)$$

The overall similarity is

$$\widehat{\rho} = \widehat{\rho}_{Cd} \times \widehat{\rho}_{Ld}. \quad (8)$$

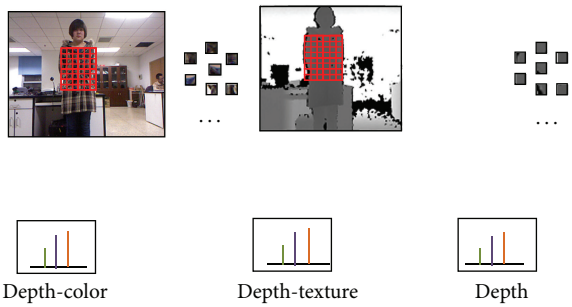


FIGURE 5: The illustration of depth information when there is occlusion.

3.5. Tracker. EKF is a set of mathematical equations providing an efficient solution for prediction problem. The algorithm is very powerful to deal with the short time occlusion problem in tracking process. However, for the person tracking system with a mobile robot, the EKF often fails to accurately predict the person because the robot and person are moving together. To deal with the problem, we present a tracker called Motion EKF combining the motion of the robot and the person:

$$\begin{aligned} X_r^{t+1} &= f\left(X_r^t, \text{control}_t\right) + R_t w_t, \\ Y_r^t &= H_t X_r^t + p_t, \end{aligned} \tag{9}$$

where $X_r = [x_r, y_r, z_r, \dot{x}_r, \dot{y}_r]$ is the state vector, (x_r, y_r, z_r) is the 3D position of the person in the robot coordinate system, $\dot{x}_r, \dot{y}_r$ are the velocity of the person in the horizontal plane, and $\text{control}_t = [v_l, v_r]$ is the control variable. p_t is the observation noise, and its covariance matrix is $R^t = \text{Cov}(p_t) = E[p_t, p_t^T] = \sigma_p^2 \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$. $Y_r^t = (x_r^t, y_r^t, z_r^t)$ is the 3D position of the target in time t. w_t is the process noise, and its covariance matrix is $Q_t = \text{Cov}(w_t) = E[w_t, w_t^T] = \sigma_w^2 \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. Consider $H_t = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{smallmatrix}\right]$.

Considering the robot's motion, the state transition function is

$$\begin{aligned} & f_t\left(x_r^t, \text{control}_t\right) \\ &= \begin{bmatrix} \left(x_r^t + \Delta t x_r^t - \Delta x_r\right)\cos\Delta\theta + \left(y_r^t + \Delta t y_r^t - \Delta y_r\right)\sin\Delta\theta \\ -\left(x_r^t + \Delta t x_r^t - \Delta x_r\right)\sin\Delta\theta + \left(y_r^t + \Delta t y_r^t - \Delta y_r\right)\cos\Delta\theta \\ z_r^t \\ x_r^t\cos\Delta\theta + y_r^t\sin\Delta\theta - v \\ -x_r^t\sin\Delta\theta + y_r^t\cos\Delta\theta \end{bmatrix}. \end{aligned} \tag{10}$$

The state equation and observation equation of the MEKF are obtained by considering the robot and person's motion. Compared with EKF, MEKF introduces the robot's trajectories to improve the robustness of the tracking. Moreover, the tracking result is a sample of the candidate set of the detector. The detector recognizes the result from the candidate set including the tracking result. The detector and tracker complement each other, which improves the ability of person detecting.

4. Person Location and Model Update

4.1. Person Location. The proposed tracking framework has been detailed in Figure 2. In the framework, the person is located by applying the detector and the tracker together. During this procedure, the depth information is fused with the color and texture information. Consequently, we obtain depth-color histograms and depth-texture histograms. Then, the person is represented with many patches' appearance models. Furthermore, to realize robustly tracking task, the detector and tracker complement each other. The process for identifying a person is as follows:

(1) Input is as follows: the depth image and color image.

(2) Get depth histograms for the depth image. Divide the depth image and color image into $N \times N$ image patches and then extract these patches' depth-color histograms and depth-texture histograms for modeling the person.

(3) For a new frame, candidate samples are obtained around the result. MEKF predicts the person's position which is treated as a sample for detector. Extract the candidate samples' depth histograms and divide the depth image and color image into $N \times N$.

(4) *For* $n = 1 : N \times N$,

  (a) extract each patch's depth-color histograms and depth-texture histograms;

  (b) compute the patch's similarity of the depth-color histograms and depth-texture histograms.

  End

(5) The patches will be in a decreasing order based on $\hat{\rho}_{Cd,n}$ and $\hat{\rho}_{Ld,n}$. Compute the similarities $\hat{\rho}_{Cd}$ and $\hat{\rho}_{Ld}$. Compute the overall similarity according to $\hat{\rho}$.

(6) Determine the occlusion problem based on the depth histograms and image pieces' similarity, and then detect the person accordingly.

(7) Output is as follows: person's position.

4.2. Model Update. Illumination changes and pose variations may result in appearance variation. To cope with this problem, an efficient update strategy should be used for adjusting to the appearance changes after detecting the person. The update strategy studies the person's appearance model according to patches' similarity in different tracking circumstances:

$$\hat{q}_{f,n,u}^t = \begin{cases} \lambda \times \hat{q}_{f,n,u}^t + (1-\lambda) \times \hat{q}_{f,n,u}^{t-1}, & \lambda > \text{th}, \\ \hat{q}_{f,n,u}^{t-1}, & \text{otherwise}, \end{cases} \tag{11}$$

where $\lambda = \hat{\rho}_{f',n}$ is the patches' appearance similarity.

Normally, the λ is the smaller similarity value of depth-color and depth-texture histograms. When one feature changes too much due to pose or illumination variations and fails to recognize the person, the λ will be determined by the

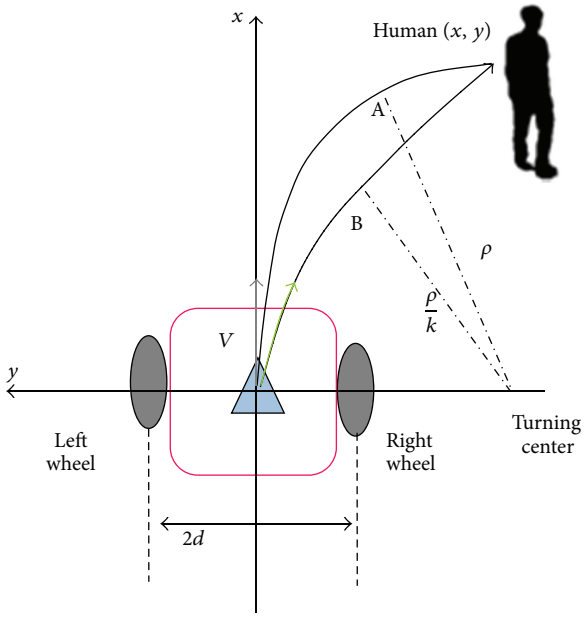


Figure 6: The path of the robot towards the person.

other feature's similarity. In such a case, both of the depth-color and depth-texture histograms are updated based on the λ. In particular, the failure appearance model changes adaptively. When there is partial occlusion, λ equals the unconcluded pieces' similarity.

5. Controller

Our goal is to design an efficient controller to drive the robot towards a given person and remain at a secure distance from him. To follow the robot smoothly and continuously, an intelligence control strategy (IGS) was presented [7], where the robot's speed and steer are controlled through introducing a turning-gain k. Using the turning gain, the robot can adaptively change the turning radius to avoid losing or crashing the person. The path of the robot towards the person is shown in Figure 6. The person's position x_r, y_r, z_r is obtained from the detector mentioned above. ρ is the turning radius of the robot to follow the person.

For path B, the velocities of the robot's wheels are computed as follows:

$$v_l = v\left(1 - \frac{2dky_r}{(x_r^2 + y_r^2)}\right),$$
$$v_r = v\left(1 + \frac{2dky_r}{(x_r^2 + y_r^2)}\right), \quad (12)$$

where v_l and v_r are the velocities of the left wheel and right wheel, respectively. x_r and y_r are the person's positions in the plane coordinate. x_r denotes the direction of the person, while y_r is for his direction.

As following evolves, the turning-gain k and the linear velocity v from the IGS keep constant. For a small turning gain, the turning radius ρ/k is large. When the direction of the person is large (the person is far away from the center of the field of view of the robot), the robot tends to lose the person. In contrast, using a large turning gain, the robot often fails to catch the person close to the center of the robot due to the small turning radius. Similarly, the robot cannot follow the person with large distance by using a small linear velocity and will hit the person due to a large linear velocity. To deal with these problems, we present a fuzzy based intelligent control strategy (FZ-IGS). The strategy includes two fuzzy controllers: a linear velocity controller and a turning-gain controller.

Table 1: The fuzzy logic for velocity controller.

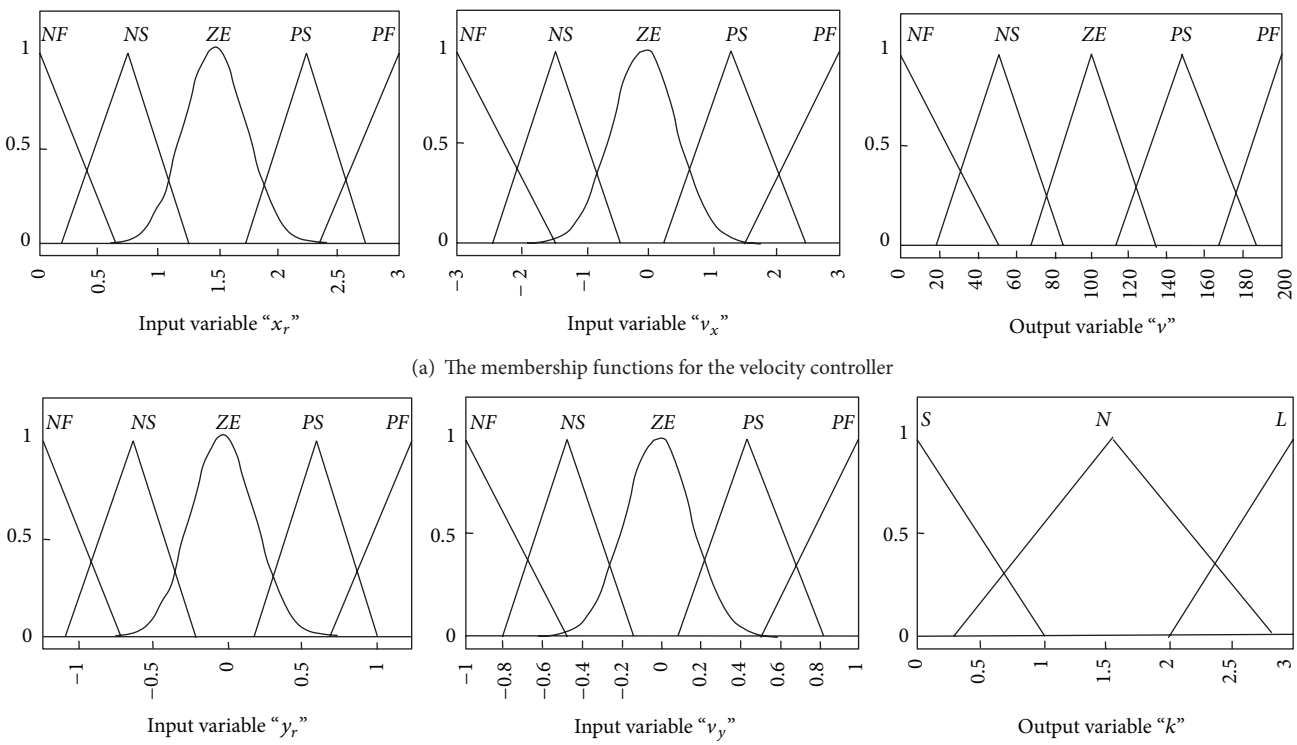
v		v_x				
		NF	*NS*	*Z*	*PS*	*PF*
x_r	*NF*	*VS*	*VS*	*S*	*S*	*Z*
	NS	*VS*	*S*	*S*	*Z*	*Z*
	Z	*VS*	*Z*	*Z*	*F*	*F*
	PS	*Z*	*F*	*VF*	*VF*	*VF*
	PF	*F*	*VF*	*VF*	*VF*	*VF*

5.1. Fuzzy Based Linear Velocity Controller. Our task is to keep the robot in a safe distance from the person while both the robot and person are moving. The distance between the robot and person varies due to their motions. In order to achieve a success track, the robot should change its linear velocity according to the distance obtained from the detector. Therefore, a fuzzy based linear velocity controller is designed to adaptively adjust the robot's velocity.

For the controller, the distance x_r and the person's vertical velocity v_x are chosen as inputs and the linear velocity v is chosen as output. For the inputs, two kinds of membership functions are used: the triangular membership function is for the large distance and velocity and the Gaussian membership function is for the small distance and velocity. For the output, we choose the triangular membership function. The domains of these parameters are $x_r \in [0, 3]$, $v_x \in [-1, 1]$, and $v \in [0, 200]$. The membership functions for these parameters are shown in Figure 7(a).

The fuzzy logic is established based on the human knowledge, which is shown in Table 1. According to the fuzzy logic, an adaptive linear velocity is obtained to drive the robot. In the case in which the distance and the speed of the person are the largest ($x_r = PF, v_x = PF$), the linear velocity will be accelerated to the maximal value ($v = PF$) for following the person as soon as possible. In contrast, if the distance and the person's velocity are very small ($x_r = NF, v_x = NF$), the robot will be slowed down to avoid hitting the person ($v = NF$). The fuzzy based controller makes the robot adapt its linear velocity according to the distance between the robot and person and the person's speed.

5.2. The Fuzzy Based Turning-Gain Controller. As following evolves, the person often wanders from the center of the robot's field. In such a case, the robot should change its turning radius in time to make sure that the person is in



(a) The membership functions for the velocity controller

(b) The membership functions for the turning-gain controller

Figure 7: The membership functions for linear velocity controller and turning-gain controller.

the center of the robot's field. To implement the task, a fuzzy based turning-gain controller is designed, where the robot's turning gain is adjusted according to the direction between the person and robot and the person's horizontal velocity.

The inputs for the fuzzy based turning-gain controller are the direction y_r and horizontal velocity of the person v_y, respectively. The output is the turning-gain k. The membership functions of the parameters for the fuzzy based turning-gain controller are the same as that of the fuzzy based linear velocity controller, shown in Figure 7(b). The domains of these parameters are $y_r \in [-1.25, 1.25]$, $v_y \in [-1, 1]$, and $k \in [0, 3]$.

The fuzzy logic is designed according to the human knowledge to determine the robot's turning-gain k. In the case in which the person moves to the left ($y_r = PF$) at positive fast speed ($v_y = PF$), the robot will turn at a very large turning gain ($k = L$) to make the human appear in the center of the robot's field again. When the person moves to the right ($y_r = NF$) at the positive speed ($v_y = PF$), the robot should turn at a normal turning gain ($k = N$) for implementing the following task. The fuzzy logic for the turning gain is shown in Table 2.

Table 2: The fuzzy logic for turning gain controller.

k		v_y				
		NF	NS	Z	PS	PF
y_r	NF	L	L	N	N	N
	NS	L	L	N	S	S
	Z	N	N	S	N	N
	PS	S	S	N	L	L
	PF	N	N	N	L	L

6. Experimental Results

Our person tracking algorithm is conducted on the Pioneer 3-DX robot.

6.1. User Is Moving but Robot Is Still. In this set of experiments, our method is compared with the color-texture based object representation algorithm [7]. These methods are evaluated on the color and depth image sequences captured from a still Kinect. The robot with the Kinect is still and a given person moves at about 1~3 m in front of the robot. In the following process, the given person moves here and there, and another person will pass by and occlude the given person. The comparison results are shown in Figure 8. The results in the first row are obtained by using the color-texture based algorithm; these in the second row are for our proposed method. When there is occlusion, the color-texture based method often fails to locate the person and loses him after occlusion. Using our method, the occlusion problem can be detected by analyzing the depth histograms and patches' similarity. Once the occlusion is detected, the method can recognize the given person by using the unoccluded patches' appearance model

Figure 8: The tracking result using CT algorithm and our method when user is moving but robot is still.

Figure 9: The tracking result using CT algorithm and our method when both of the user and the robot are moving.

represented by depth-color and depth-texture histograms. The results show that the patches based algorithm is of benefit to the occlusion problem. Furthermore, taking advantage of depth information, the person representation makes the foreground-background segmentation much easier.

6.2. Both the User and Robot Are Moving. In this section, our method is evaluated on a moving robot. As tracking evolves, there are occlusion, turning, appearance changes, and motion of both the robot and the given person. The tracking results are shown in Figure 9. The method tracks the person by adopting a patches based multicues detector and a MEKF tracker. In the case in which there is partial occlusion, the person is successfully detected by performing our method. When the person is fully occluded, the MEKF predicts the position of the person. Furthermore, an update strategy is adopted for updating the appearance representation in the tracking process. The experiment results illuminate that our method performs well in case of occlusion and appearance variations.

6.3. Robot Following Based on FZ-IGS. In this section, the performance of the presented FZ-IGS is evaluated. The given person's 3D position (x_r, x_y, x_z) was obtained from the data provided by the Kinect and was sent to the robot's controllers. The fuzzy based velocity controller changes the linear velocity according to the fuzzy logic. When there are variations in terms of distance between the robot and person (x_r) or the person's vertical speed (v_x), the linear velocity will accordingly change to make sure that the person is in a safe distance from the robot. Similarly, the turning-gain controller determines the turning gain according to the person's direction (x_y) and his speed (v_y) to keep the person in the center of the field of view of the robot. The paths for a person following robot are illustrated in Figure 10. For Figure 10(a), the red symbols "+" denote the path of the person, while the blue symbols "∘" are for the path of the robot. "0" is the start point of the person. In the beginning, the robot is still and the person is moving. When the distance between the person and the robot is larger than the safe distance (at about the point $x = 2000$, $y = 0$), the robot starts to follow the person. The results show that the robot can follow the person in a safe distance and keep him at the center of its FOV. In the case in which the distance or direction changes, the robot can vary its linear velocities and the turning gain to make sure that the robot can follow the robot stably. Figures 10(b) and 10(c) show the vertical distance and horizontal distance between the robot and the target according to time t, respectively. The results show that our method can guarantee that the robot tracks the person in a safe distance.

7. Conclusion

In this paper, we developed a new person tracking algorithm for a mobile robot. The paper exhibited four contributions. The first contribution concerned the person representation algorithm based on the fusion of multicues including depth-color histograms and depth-texture histograms. The color and texture information complement each other which improves the appearance's discrimination ability. The depth information easily discriminates the person from the background. The second contribution concerned patches based detection algorithm which divided the person into many patches. It could handle the partial occlusion problem by analyzing the unoccluded patches' similarity. The third contribution concerned the tracker MEKF which considers the motion of the robot and person. The fourth contribution

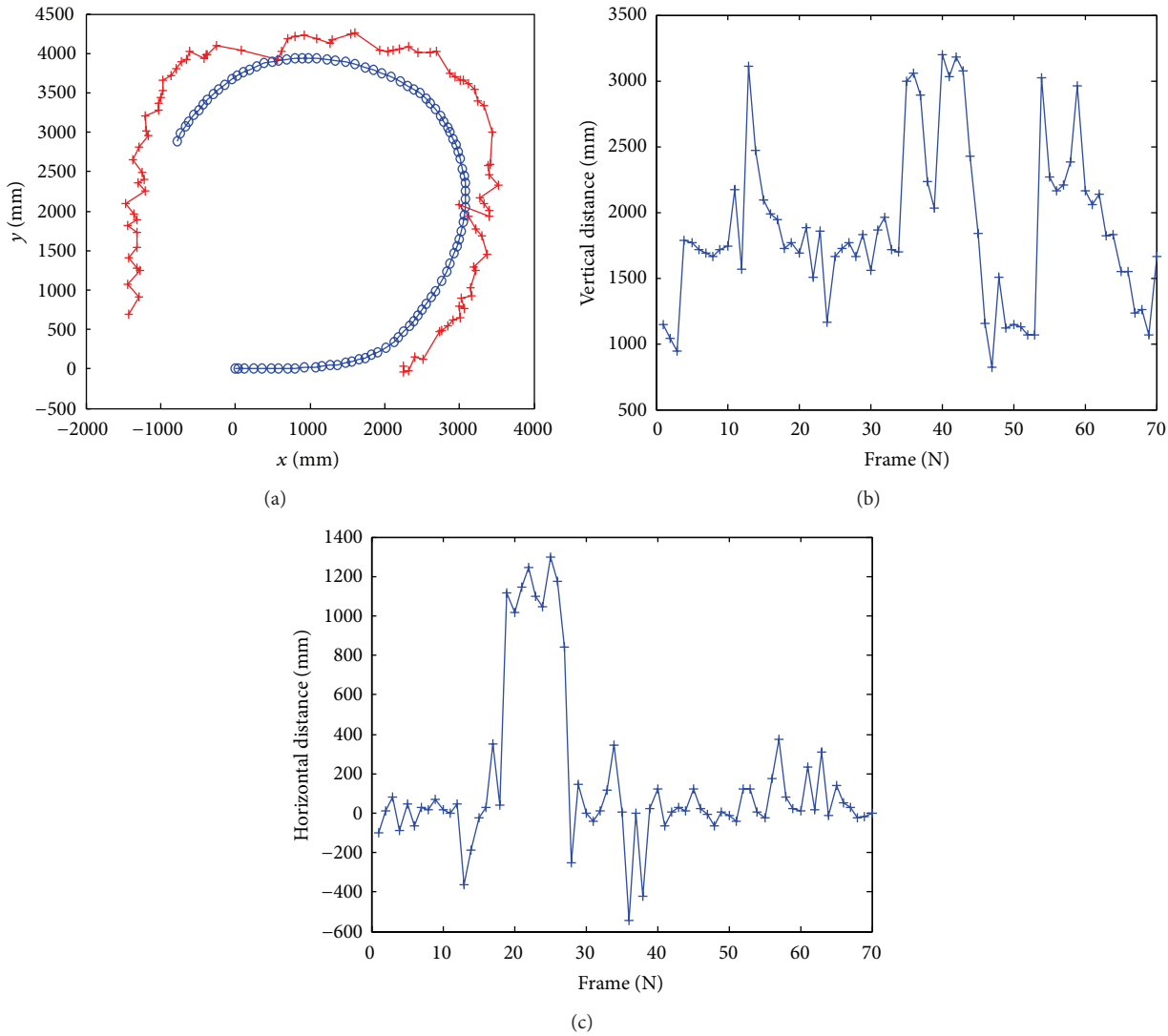


Figure 10: The robots path for following the target in the Lab.

concerned the fuzzy based intelligent controllers (FZ-IGS) which can adaptively change the linear velocity and turning-gain according to the person's positions obtained from the detector. The experimental results have demonstrated that the proposed method is able to track a person robustly and accurately. In the future, we will study the obstacle avoidance method in the tracking process.

Conflict of Interests

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

Acknowledgments

The research work is supported by National Natural Science Foundation (61175087), the Youth Foundation of Hebei Education Department, QN2014170, and Hebei province Science and Technology Support Program, 14275601D.

References

[1] M. Bansal, B. Matei, H. Sawhney, S.-H. Jung, and J. Eledath, "Pedestrian detection with depth-guided structure labeling," in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops '09)*, pp. 31–38, Kyoto, Japan, October 2009.

[2] A. H. Mazinan and A. Amir-Latifi, "Applying mean shift, motion information and Kalman filtering approaches to object tracking," *ISA Transactions*, vol. 51, no. 3, pp. 485–497, 2012.

[3] Z. Zivkovic and B. Kröse, "An EM-like algorithm for color-histogram-based object tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. I-798–I-803, IEEE, July 2004.

[4] J. F. Ning, L. Zhang, D. Zhang, and C. K. Wu, "Scale and orientation adaptive mean shift tracking," *IET Computer Vision*, vol. 6, no. 1, pp. 52–61, 2012.

[5] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust object tracking using joint colour texture histogram," *International Journal of*

Pattern Recognition and Artificial Intelligence, vol. 23, no. 7, pp. 1245–1263, 2009.

[6] O. Zoidi, N. Nikolaidis, A. Tefas, and I. Pitas, "Stereo object tracking with fusion of texture, color and disparity information," *Signal Processing: Image Communication*, vol. 29, no. 5, pp. 573–589, 2014.

[7] S. Jia, S. Wang, L. Wang, and X. Li, "Robust human detecting and tracking using varying scale template matching," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA'12)*, pp. 25–30, June 2012.

[8] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, vol. 30, no. 3, pp. 217–226, 2012.

[9] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG '11)*, pp. 700–706, March 2011.

[10] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by Kinect," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '11)*, pp. 15–22, IEEE, Colorado Springs, Colo, USA, June 2011.

[11] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, June 2006.

[12] S. M. S. Nejhum, J. Ho, and M.-H. Yang, "Visual tracking with histograms and articulating blocks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, 2008.

[13] M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[14] J. Kwon and K. M. Lee, "Highly nonrigid object tracking via patch-based dynamic appearance modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2427–2441, 2013.

[15] N. Ouadah, V. Cadenat, F. Lerasle, M. Hamerlain, T. Germa, and F. Boudjema, "Multi-sensor-based control strategy for initiating and maintaining interaction between a robot and a human," *Advanced Robotics*, vol. 25, no. 9-10, pp. 1249–1270, 2011.

[16] M. Tarokh, P. Merloti, J. Duddy, and M. Lee, "Vision based robotic person following under lighting variations," in *Proceedings of the 3rd International Conference on Sensing Technology (ICST '08)*, pp. 147–152, December 2008.

[17] O. Azouaoui, N. Ouadah, I. Mansour, and A. Semani, "Fuzzy motion-based control for a bi-steerable mobile robot navigation," in *Proceedings of the 6th International Symposium on Mechatronics and Its Applications (ISMA '09)*, March 2009.

[18] L. Cao, C. Wang, and J. Li, "Robust depth-based object tracking from a moving binocular camera," *Signal Processing*, vol. 112, pp. 154–161, 2015.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 869175, 15 pages
http://dx.doi.org/10.1155/2015/869175

Research Article

A Control Strategy with Tactile Perception Feedback for EMG Prosthetic Hand

Changcheng Wu, Aiguo Song, Yun Ling, Nan Wang, and Lei Tian

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Aiguo Song; a.g.song@seu.edu.cn

Received 6 September 2014; Revised 13 January 2015; Accepted 18 January 2015

Academic Editor: Lei Wang

To improve the control effectiveness and make the prosthetic hand not only controllable but also perceivable, an EMG prosthetic hand control strategy was proposed in this paper. The control strategy consists of EMG self-learning motion recognition, backstepping controller with stiffness fuzzy observation, and force tactile representation. EMG self-learning motion recognition is used to reduce the influence on EMG signals caused by the uncertainty of the contacting position of the EMG sensors. Backstepping controller with stiffness fuzzy observation is used to realize the position control and grasp force control. Velocity proportional control in free space and grasp force tracking control in restricted space can be realized by the same controller. The force tactile representation helps the user perceive the states of the prosthetic hand. Several experiments were implemented to verify the effect of the proposed control strategy. The results indicate that the proposed strategy has effectiveness. During the experiments, the comments of the participants show that the proposed strategy is a better choice for amputees because of the improved controllability and perceptibility.

1. Introduction

Prosthetic hands are of great importance to the upper limbs amputees which can help them to complete some manipulations such as grasping. How to make the prosthetic hand grasp objects by following inclinations of amputees is a meaningful research direction. Scholars in this filed have done a lot of research work and made some important development.

At present, there are some kinds of prosthetic hands: decorative prosthetic hand (for the purpose of decoration, without function), switches controlled prosthetic hand, electromyography (EMG) signal controlled prosthetic hand, and so forth. A typical control scheme of EMG prosthetic hand is shown in Figure 1 [1]. EMG_1 and EMG_2 are EMG signals acquired from a pair of antagonistic muscles, K_E is the scale factor, F_d is the expected grasp force, and F_n is the grasp force measured by force sensors. K_n is the feedback gain and K_p is the proportional gain for the force error. This control mode is popular due to its simple operation and in accordance with the operation habits of natural hand. However, there are still some problems to be solved. Firstly, this control mode does not fully consider the influence of EMG signals on control strategy: the EMG signals measured from different persons are different, and the EMG signals measured from different state of the same individual may be different. Secondly, this control mode lacks perceptibility: the user cannot perceive the grasp force of the prosthetic hand when grasping an object.

EMG motion pattern recognition is a basic technique of EMG prosthetic hand; a great number of recognition methods have been developed, such as time-domain method and frequency-domain method [2]. Parameters of ARMA model and Kalman filter were adopted as character vectors to identify the movement patterns [3]. Khoshaba et al. used the integral absolute value of the EMG signals to recognize the movement patterns [4]. Autoregressive (AR) model, power spectrum, wavelet coefficients, neural network, and some signal processing methods were also used [5]. However, due to the difference of the sticking positions of EMG sensors and the difference of temperature and humidity of the environment, the EMG signals measured from different persons or different states of the same individual may lead to different results. This kind of phenomenon may affect

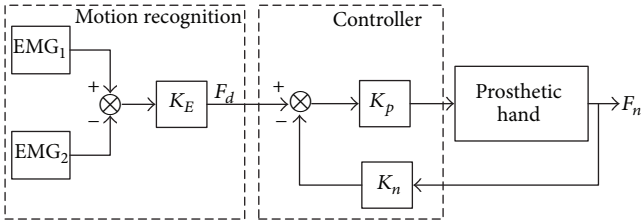


Figure 1: A typical control scheme of the EMG prosthetic hand.

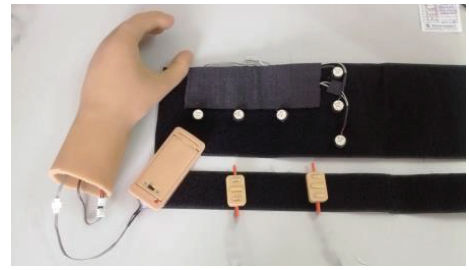


Figure 2: MPH-III prosthetic hand.

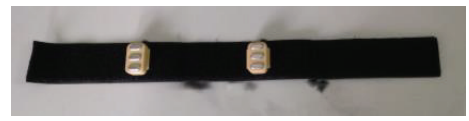


Figure 3: EMGs acquisition device.

the accuracy of the movement pattern recognition, which may cause descending of the control effect.

The design of the controller is another important part of the EMG prosthetic hand; most of the control strategies for the prosthetic hands aimed at the controllability, such as accuracy force tracking control and speed control of opening and closing. In order to get a satisfying grasping performance, Fassih et al. proposed a control strategy based on defining virtual spring-damper model between two finger tips and damping force at each finger joint [6]. Chen et al. focused on hybrid of soft computing technique of adaptive neuron-fuzzy inference system (ANFIS) and hard computing technique of adaptive control for a two-dimensional movement with thumb and index prosthetic hand [7]. A hybrid sliding mode-backstepping (HSMBS) parallel force-velocity controller was proposed to improve the control effect of powered prosthetic hands by Engeberg and Meek [8]. According to the tactile feedback and visual feedback, when grasping an object, natural hand can adjust the grasp force in time. But for prosthetic hand which has no tactile feedback to the user, it is hard to obtain expected control effect. So a prosthetic hand which is not only controllable but also appreciable is probably a good choice for amputees. The controllability of the prosthetic hand just functionally assists amputees to complete some simple actions such as grasping. However, the perceptibility of the prosthetic hand considers more about the amputee himself. It not only satisfies the functional needs of the hands but also meets the psychological requirements. In a view of the perceptibility of the prosthetic hand, amputees get more information when using it. It shows a faster acceptance for amputees when using the hand for a long time, and this phantom limb feeling may also improve the control effect.

This paper describes a control strategy for the EMG prosthetic hand, which mainly includes EMG self-learning recognition, backstepping controller with stiffness fuzzy observation, and grasp force tactile representation. EMG self-learning recognition aims to reduce the influence on EMG signals caused by the uncertainty of the contacting positions of the EMG sensors. Backstepping controller with stiffness fuzzy observation is used to realize the position control and grasp force control. The grasp force tactile representation aims to improve the proprioception of the prosthetic hand, which can improve the control effect. Finally, experiments were implemented to verify the proposed control strategy.

2. System Components

The MPH-III prosthetic hand, which is designed by Robot Sensor and Control Lab in Southeast University, is used in this paper. The MPH-III consists of three components: prosthetic hand, EMGs acquisition device, and tactile representation device (see Figure 2). The lithium batteries are used as power source for all components.

2.1. Prosthetic Hand. The MPH-III is a one-DOF (degree of freedom) prosthetic hand, which is equipped with three force sensors (to measure the grasp force) and a position sensor (encoder). Wearing silicone glove, the MPH-III looks like a natural hand. The core of the control system is a single chip microcomputer (C8051F320, Silicon Laboratories) [9]. The control board has a USB interface on which data can be exchanged between prosthetic hand's control board and computer.

2.2. EMGs Acquisition Device. Two EMG sensors are adopted to acquire EMG signals. A 10-bit A/D converter is used to digitize signals (sampling ratio is 1 kHz). A Bluetooth module is utilized to transmit the EMG signals to the controller of prosthetic hand. All the components of the EMGs acquisition device are fixed on a ribbon (see Figure 3) to make it convenient for users to wear, and the positions of the EMG sensors on the ribbon are adjustable because the detecting positions for different users are different.

2.3. Tactile Representation Device. The tactile representation device (TRD) consists of six vibration motors, which are controlled by an electronic system equipped with a Bluetooth module. The vibration motors are fixed on a ribbon (see Figure 4). When the ribbon is worn on the upper arm, the distribution of vibration motors is shown in Figure 5. These vibration motors may generate stimulation on the skin of the amputee. The TRD receives the force information of the prosthetic hand from the control module via Bluetooth module. Then command is generated to control the vibration

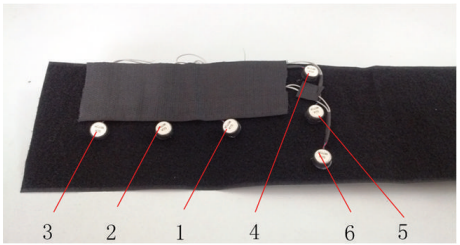


Figure 4: Tactile representation device.

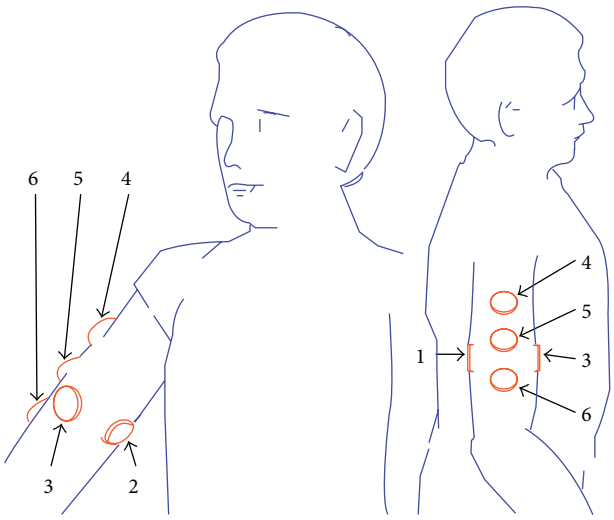


Figure 5: The distribution of the vibration motors.

motors orderly, so that the user can perceive the force states of the prosthetic hand.

3. Design of the Control Strategy

The designed control strategy is based on manipulation patterns of the natural hand. User plays a dominant role in the system. When she/he intends to operate the prosthetic hand, a control command will be generated from the brain and transmitted to motor nerves through spinal cord. The motor nerves control the muscle movement. And the EMG signal will be generated on the surface of skin at that time. These EMG signals can be used to control the prosthetic hand after processing. Processing these EMG signals probably involves amplifying, filtering, acquiring, feature extracting, and motion identifying. The force sensors installed in fingers of the prosthetic hand are used to detect the grasp force. On one hand, the grasp force information feeds back to the controller; on the other hand it feeds back to the user through a certain tactile feedback type. According to the specialty of the object, users can combine visual and tactile message and adjust the control strategy in time. In this way, three closed loops are achieved to control the prosthetic hand: the first one is from user to prosthetic hand then back to the user through tactile feedback device, the second one is from controller to prosthetic hand then back to the controller, and the last one is from the user to prosthetic hand then back to the user through user's eyes. The functional scheme of this control strategy is shown in the Figure 6.

3.1. EMG Self-Learning Recognition. The accuracy of the pattern recognition of EMG signals is directly related to the control effect of the prosthetic hand. The surface EMG signal can illustrate the activity of skeletal muscles, and its amplitude ranges from less than 50 μV to 30 mV, and the frequency range is from dozens to hundreds Hz, depending on the muscle under observed.

Because the EMG signal strengths of different users are different and due to some other factors, an EMG self-learning recognition method is proposed as shown in Figure 9. Before the pattern recognition, EMG signals are processed as shown in Figure 7.

EMG sensors are attached to the surface of the muscle to acquire the EMG signals. Figures 8(a), 8(b), and 8(c) show the amplified EMG signals, the shaped EMG signals, and the filtered EMG signals, respectively.

In Figure 9, EMG_1 and EMG_2 are the EMG signals which are preprocessed as shown in Figure 7. The values of EMG_1 and EMG_2 are between 0 V and 3.3 V. The EMG learner is designed to record and update the minimum and maximum values of the EMG signals. A moving window is adopted, and the principles of the recording and updating are as follows:

Step 1. Calculate the average value (Ave) of the data in moving time window:

$$\mathrm{Ave} = \frac{1}{N}\sum_{n=0}^{N-1} \mathrm{EMG}_1(t-n), \tag{1}$$

where N is the length of the time window. $\mathrm{EMG}_1(t)$ represents the current data of the EMG signal and $\mathrm{EMG}_1(t-n)$ represents the previous nth data of the EMG signal.

Step 2. Update the data:

$$\begin{aligned} \mathrm{Max}_{E1} &= \begin{cases} \mathrm{Max}_{E1}, & \mathrm{Max}_{E1} \geq \mathrm{Ave} \\ \mathrm{Ave}, & \mathrm{Max}_{E1} < \mathrm{Ave} \end{cases} \\ \mathrm{Min}_{E1} &= \begin{cases} \mathrm{Min}_{E1}, & \mathrm{Min}_{E1} \leq \mathrm{Ave} \\ \mathrm{Ave}, & \mathrm{Min}_{E1} > \mathrm{Ave}, \end{cases} \end{aligned} \tag{2}$$

where Max_{E1} and Min_{E1} are the maximum and minimum values of the EMG_1, respectively. The initial value of Max_{E1} is set to 0, and the initial value of Min_{E1} is set to 3.3.

The maximum and minimum values of the EMG_2, Max_{E2}, and Min_{E2} are updated by using the same method.

Adjustable factors (K_{E1}, K_{E2}) are defined as follows:

$$\begin{aligned} K_{E1} &= \frac{1}{\mathrm{Max}_{E1} - \mathrm{Min}_{E1}}, \\ K_{E2} &= \frac{1}{\mathrm{Max}_{E2} - \mathrm{Min}_{E2}}. \end{aligned} \tag{3}$$

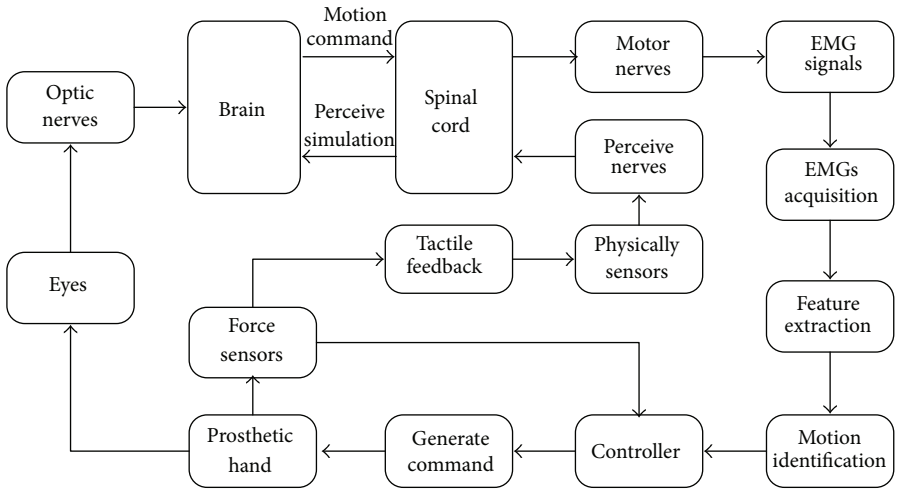


Figure 6: Control strategy functional scheme.

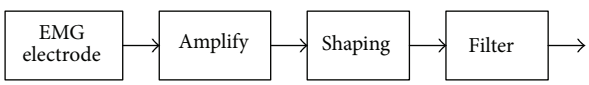


Figure 7: Preprocessing scheme of the EMG signals.

The output, F_d, is determined by EMG_1 and EMG_2, which is as follows:

$$F_d = \left[\left(\mathrm{EMG}_1 - \mathrm{Min}_{E1}\right) \times K_{E1} - \left(\mathrm{EMG}_2 - \mathrm{Min}_{E2}\right) \times K_{E2}\right] \times K_E, \tag{4}$$

where K_E is the scale factor.

In free space, F_d reflects the closing or opening speed of the prosthetic hand, and in restricted space, F_d reflects the grasp force.

With the help of this recognition method, the influence of the diversity of the EMG signals on the accuracy of the pattern recognition is reduced.

3.2. Backstepping Controller with Stiffness Fuzzy Observation. To realize the position control and grasp force control, a backstepping controller with stiffness fuzzy observation (BCSFO) is designed in this paper. The designed controller is shown in Figure 10. The input signal, F_d, is the output of the motion recognizer. F_n is the grasp force measured by force sensor which is attached to the prosthetic hand's finger, K_n is the scaling factor, K_{nd} is the differential scaling factor, and u is the voltage applied to motor. x_1 and x_2 are, respectively, the position and velocity of the prosthetic hand's finger. k is object's stiffness and the stiffness is defined as

$$k = \frac{F_n}{x_0 - x_1}\,(N/\circ), \tag{5}$$

where x_0 is the original size of the object and it is the position of the prosthetic hand's finger when the object and finger contact for the first time.

In free space, the output of the planner is as follows:

$$\theta_d = \theta_0 + \int \left(F_d - K_n F_n - K_{nd}\frac{d}{dt}F_n\right) dt = \theta_0 + \int F_d dt, \tag{6}$$

where θ_0 is the position when prosthetic hand's finger changes from restricted space to free space.

In restricted space, the stiffness of the object and the deformation of the structure may affect the relationship between the grasp force of the prosthetic hand and the angle of the motor rotation. Since the range of the designed grasp force is relatively small (0~30 N), the influence of the structure deformation is ignored, and the output of the planner in restricted is as follows:

$$\theta_d = \theta_n + \frac{F_d - \left(K_n F_n + K_{nd}\left(d/dt\right) F_n\right)}{k}. \tag{7}$$

The system model of prosthetic hand is selected as follows:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= m\left(x_1, x_2\right) + nu, \\ m\left(x_1, x_2\right) &= -\frac{B}{J}x_2 - \frac{D}{J}, \end{aligned} \tag{8}$$

where B, J, and D are, respectively, the inertia, damping, and unknown nonlinear damping of the system. u is the output of the system, and it is the control voltage of the motor.

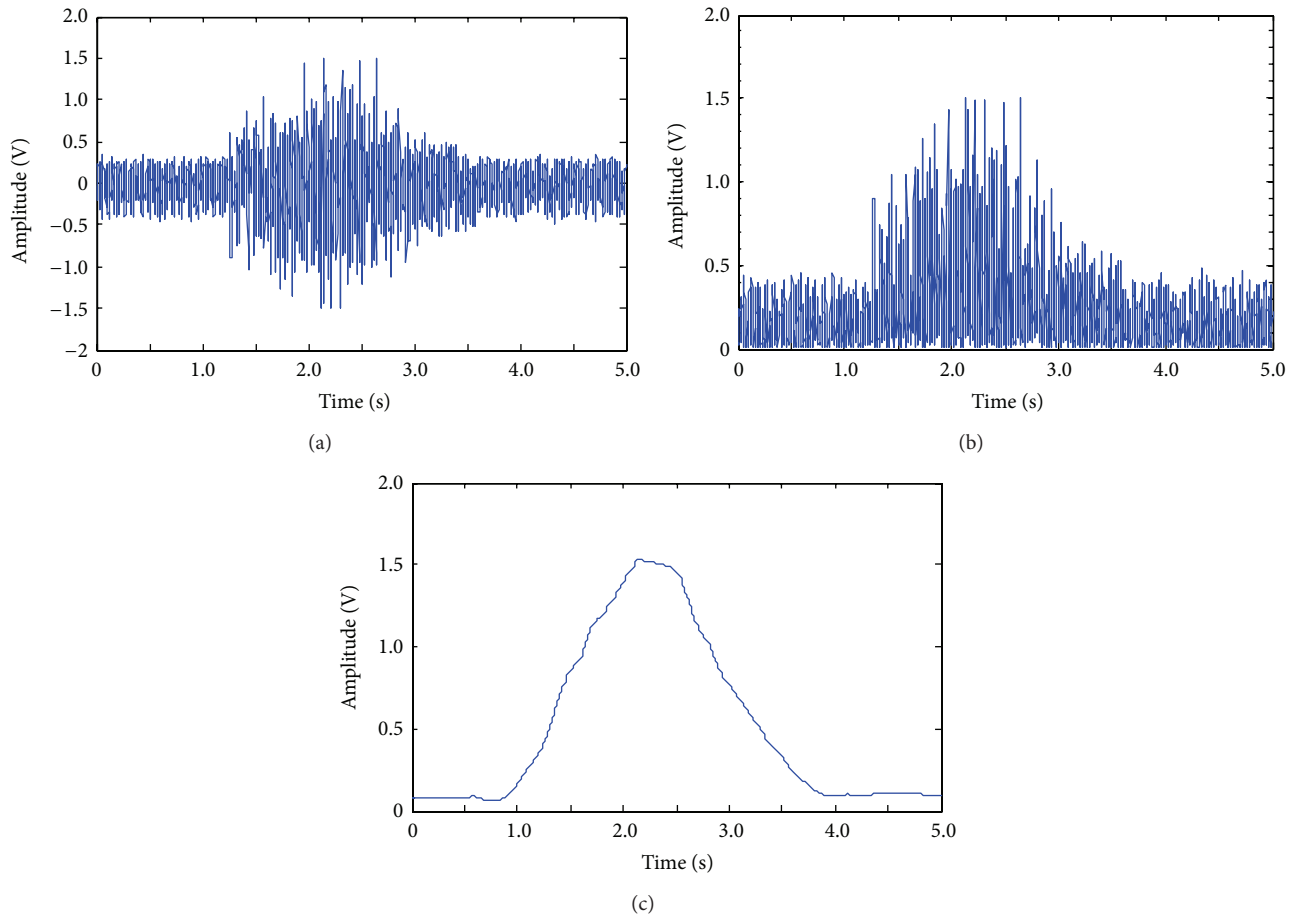


Figure 8: Amplified, shaped, and filtered EMG signals.

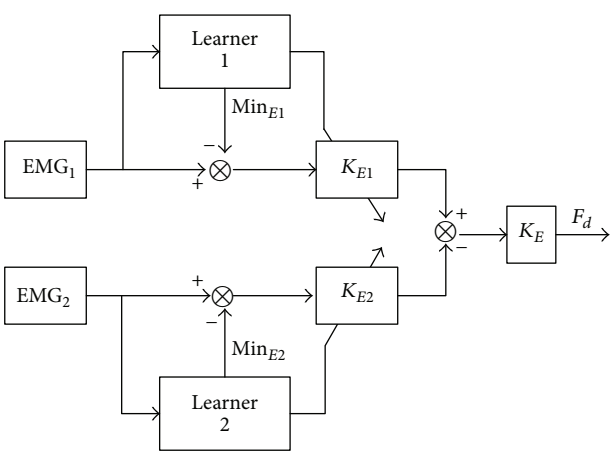


Figure 9: EMG signal self-learning recognition scheme.

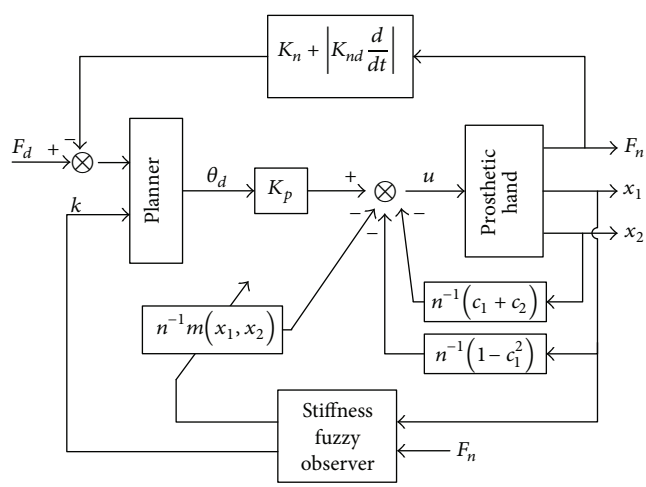


Figure 10: The block diagram of the backstepping controller with stiffness fuzzy observation.

Two error subsystems are defined as

$$Z_1 = x_1,$$
$$Z_2 = x_2 - \alpha_1(x_1), \tag{9}$$

where $\alpha_1(x_1)$ is a virtual control variable (i.e., the estimate of x_2):

$$\dot{Z}_1 = \dot{x}_1 = x_2. \tag{10}$$

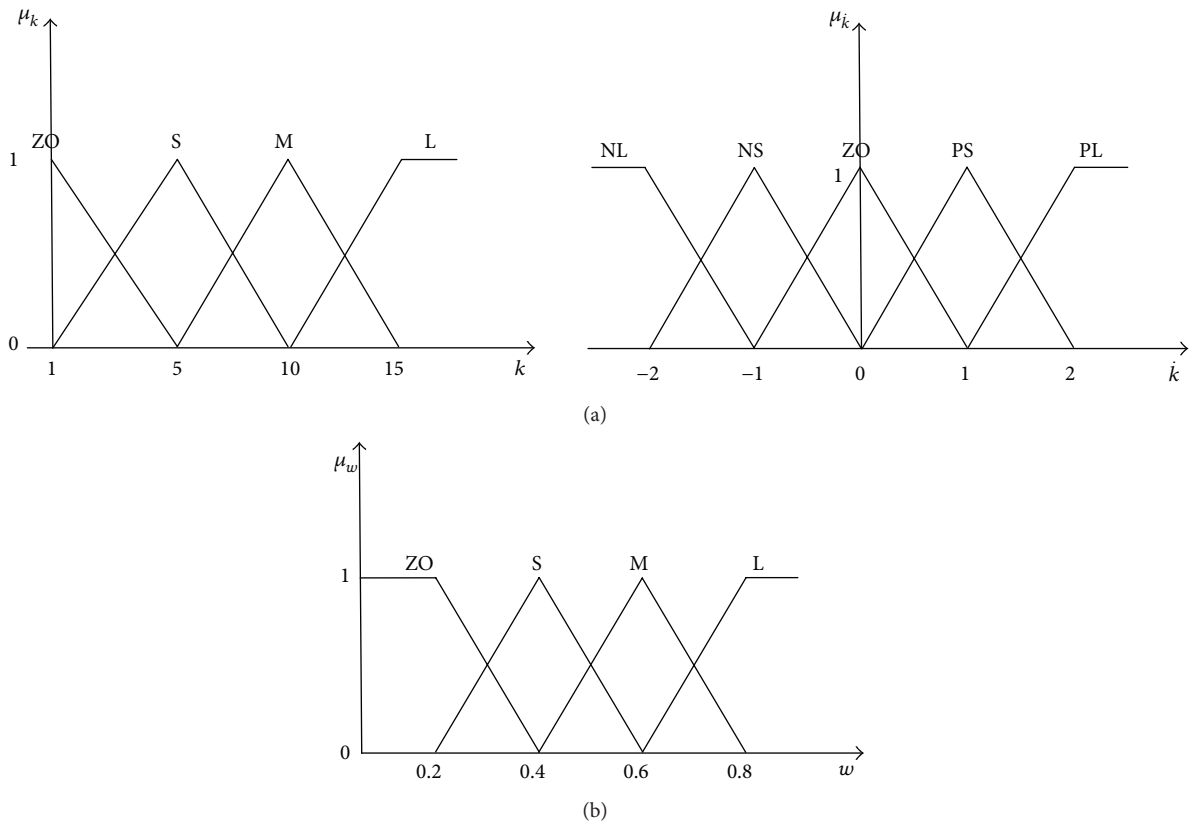


Figure 11: Membership functions for fuzzy reasoning. (a) Input member functions. (b) Output member functions.

Take $\alpha_1(x_1) = -c_1 Z_1$, $c_1 > 0$; then

$$\dot{Z}_1 = Z_2 + \alpha_1(x_1) = Z_2 - c_1 Z_1. \tag{11}$$

The Lyapunov function of the first error subsystem is defined as

$$\begin{aligned} V_1 &= \frac{1}{2} Z_1^2, \\ \dot{V}_1 &= Z_1 \dot{Z}_1 = -c_1 Z_1^2 + Z_1 Z_2, \\ \dot{Z}_2 &= \dot{x}_2 - \dot{\alpha}_1(x_1) \\ &= m(x_1, x_2) + nu + c_1 \dot{Z}_1 \\ &= m(x_1, x_2) + nu - c_1^2 Z_1 + c_1 Z_2. \end{aligned} \tag{12}$$

The Lyapunov function of the second error subsystem is defined as

$$\begin{aligned} V_2 &= V_1 + \frac{1}{2} Z_2^2, \\ \dot{V}_2 &= \dot{V}_1 + Z_2 \dot{Z}_2 \\ &= -c_1 Z_1^2 + Z_1 Z_2 + Z_2 \left[m(x_1, x_2) + nu + c_1 \dot{Z}_1 \right] \\ &= -c_1 Z_1^2 + Z_1 Z_2 + Z_2 \left[m(x_1, x_2) + nu - c_1^2 Z_1 + c_1 Z_2 \right]. \end{aligned} \tag{13}$$

Take the control law for u as follows:

$$u = \frac{1}{n} \left[-m(x_1, x_2) - \left(1 - c_1^2\right) Z_1 - (c_2 + c_1) Z_2 \right]. \tag{14}$$

Then

$$\dot{V}_2 = -c_1 Z_1^2 - c_2 Z_2^2 \le 0, \quad c_1 > 0, \ c_2 > 0. \tag{15}$$

By the Lyapunov stability theory, the designed control system can reach a steady state in a limited time, so the system has the stability.

In order to grasp objects with different stiffness stably, a stiffness fuzzy observer is designed to adjust $m(x_1, x_2)$. The employed fuzzy logic reasoning has double inputs (k and $\dot{k}$)

Table 1: Fuzzy reasoning rules.

w		k			
		ZO	S	M	L
$\dot{k}$	NL	ZO	ZO	S	M
	NS	ZO	S	S	M
	ZO	ZO	S	M	L
	PS	S	S	M	L
	PL	S	M	L	L

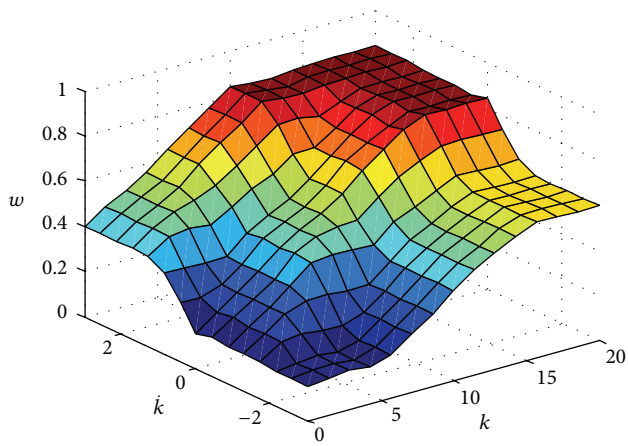


Figure 12: Input–output relationship surface map for fuzzy reasoning.

and a single output (w). k and $\dot{k}$ are, respectively, the stiffness and the stiffness' derivative. $m(x_1, x_2)$ is adjusted as follows:

$$m(x_1, x_2) = -(1 + w)\frac{B}{J}x_2 - \frac{D}{J}. \tag{16}$$

During fuzzification and defuzzification, the inputs and the output types are defined as several fuzzy sets with trigonometry/trapezoidal membership functions as shown in Figure 11. The fuzzy reasoning rules for w are shown in Table 1. Figure 12 shows the overall input–output relationship of the fuzzy logic reasoning.

3.3. Tactile Representation System. There are lots of neurons distributed on the skin all over the body. The mechanism is called tactile perception. These neurons can receive the information (temperature, humidity, pain, pressure, vibration, etc.) outside the body. One of the most common phenomena of tactile perception is that when natural hand touches an object, the characteristics of the object such as the shape and surface roughness can be felt. Tactile perception is a way to obtain information. It involves human physiology, tactile physiology, tactile phenomenon, and so forth [10, 11].

When working in this, tactile representation is important. It is important to transmit grasp force and slide information back to the user.

3.3.1. Grasp Force Detection. In order to achieve a more comprehensive and more accurate measurement of the grasp force, several force sensors are developed [12, 13]. Figure 13(a) shows the thumb of the prosthetic hand. A force sensor (FSS SMT Series, Honeywell) is installed on the tip of the finger; the other two half bridges (Wheatstone half bridge) are separately fixed on the middle and the root of the thumb. This kind of distribution is utilized because the force points on the finger may be different when user grasps objects.

The sensitivity of the force sensor is 12.2 mV/N. The force, F_1, applied to area 1 is as follows:

$$F_1 = \frac{u_F}{12.2\,\text{mV}}, \tag{17}$$

where u_F is the output of the force sensor.

Four strain gauges (sg1, sg2, sg3, and sg4) which have the same properties (material, size, strain coefficient, etc.) are attached to the thumb of the prosthetic hand as shown in Figure 13. R_1, R_2, R_3, and R_4 represent the resistance of these strain gauges, respectively. And R_1, R_2, R_3, and R_4 are of the same value.

The thumb of the prosthetic hand is made of aluminum. There is a deformation in the thumb when the force is applied to it. The deformation of the thumb will lead to the corresponding deformation of the strain gauges. And the deformation of the strain gauges will lead to the change of the resistances' value of these strain gauges.

Take sg1 and sg2; for example, tensile deformation and compressive deformation occurred in sg1 and sg2, respectively, when the force is applied to the thumb. Tensile deformation of sg1 results in the increasing of the resistance' value of sg1. On the contrary, compressive deformation of sg2 results in the decreasing of the resistance' value of sg2. The variations of the resistance' value of sg1 and sg2 are considered to be the same for the reason that the deformation is small and sg1 and sg2 have the same properties.

sg1 and sg2 are connected in the measurement circuit as shown in Figure 13(b). sg1, sg2, and two additional resistances (R_{a1}, R_{a2}) constitute a Wheatstone bridge. The output of the circuit, U_{g1}, is as follows:

$$\begin{aligned} U_{g1} &= \frac{R_2 - \Delta R}{(R_1 + \Delta R) + (R_2 - \Delta R)} \times E - \frac{R}{R + R} \times E \\ &= -\frac{\Delta R}{2 \times R_1} \times E, \end{aligned} \tag{18}$$

where E is the power voltage supplied to the circuit, ΔR is the change of resistance of sg1 and sg2, and R is the value of R_{a1} and R_{a2}.

Because the force, F_2, applied to area 2 (as shown in Figure 13) is proportional to the deformation of the strain gauges, it can be obtained by measuring the output of the circuit (U_{g1}, shown in Figure 13(b)).

The force, F_3, applied to area 3 can be obtained by using the same method.

When the prosthetic hand contacts the object in areas 1, 2, and 3, the grasp forces are measured by force sensor, half bridge 1, and half bridge 2, respectively.

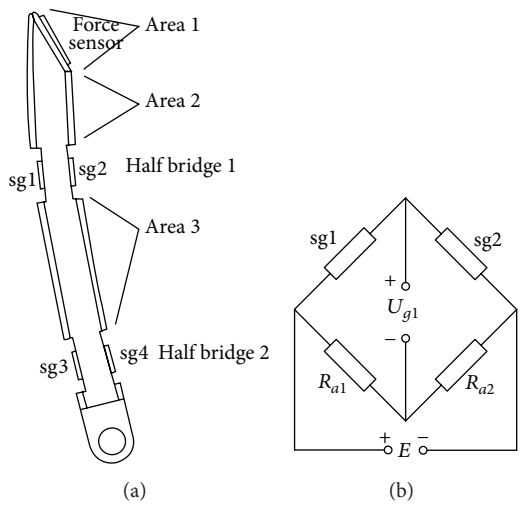


Figure 13: The thumb of the prosthetic hand and the sensor measure circuit.

3.3.2. Sliding State Detection. A polyvinylidene fluoride (PVDF) piezoelectric film is often used for tactile sensor design to detect the sliding for its excellent dynamic characteristics and physical characteristics, such as light quality, soft, large contact resistance, and plasticity [14, 15]. In this paper, a PVDF piezoelectric film is attached to the surface of the silicone glove of the prosthetic hand. The signal in Figure 14(a) shows a sliding during a grasp operation. In this figure, T indicates the time span of a sliding process.

To reduce the influence of noise on sliding detection, the filtering process is implemented. In addition, a small threshold is subtracted. Figure 14(b) shows the sliding signals after these two processes. Compared with the sliding signals, temperature is a slow change variable. The influence of the temperature is ignored when using PVDF to detect the slide state.

According to the signals shown in Figure 14 and the characteristics of the PVDF, the number of zero-crossing per time unit is adopted to indicate the sliding situation.

3.3.3. Tactile Representation. The main tactile representation techniques are pneumatic stimulation, vibration stimulation, functional neuromuscular stimulation, thimble stimulation, thermal stimulation, and so forth [16]. The vibration stimulation is adopted because it is convenient to use and does not cause damage to human body. The vibration coding patterns, including vibration frequency, amplitude, duration, rhythm, and order [17, 18], affect the accuracy of the tactile perception of the user directly. However, due to the existence of the tactile illusion phenomenon, an efficient vibration coding pattern must be established. The tactile illusion is a kind of phenomenon in which tactile perceptions do not match the objective stimulation. Many kinds of tactile illusion phenomena have already been discovered, such as phantom sensation and apparent movement. The reasonable utilization of these phenomena may contribute to realization of the tactile representation.

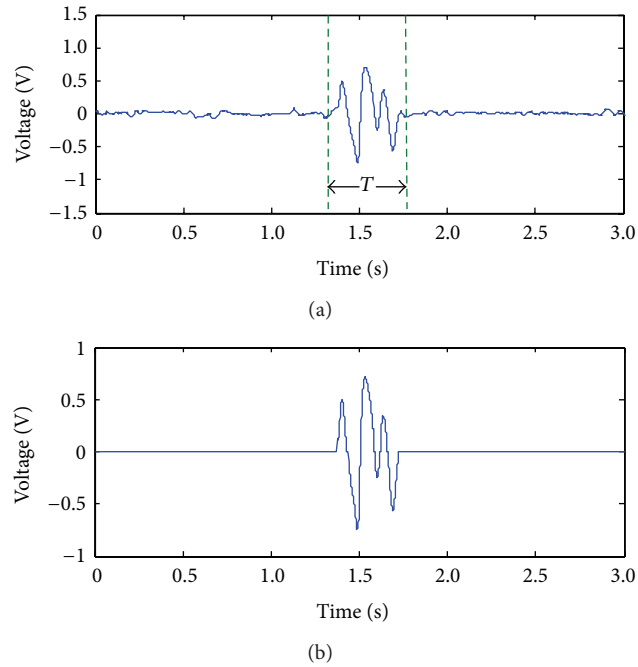


Figure 14: Sliding signals.

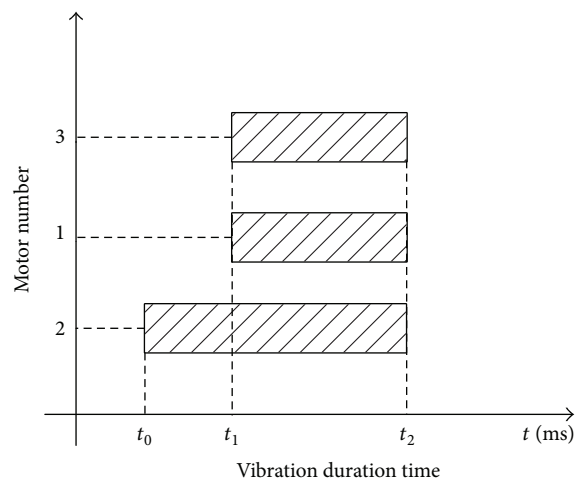


Figure 15: The coding pattern of the grasp force tactile representation.

A coding pattern (vibration coding) of grasp force is designed. As shown in Figure 15, when the grasp force is detected, motor number 2 begins to vibrate, and then motor number 1 and motor number 3 begin to vibrate. t_0 is the beginning time of motor number 2, t_1 is the beginning time of motors number 1 and number 3, and t_2 is the ending time of all these motors. t_0 is the time of grasp force being detected as well, and t_2 shows the time when the grasp force reduces to zero. The interval between t_0 and t_1 is 300 ms. The grasp force measured by the sensor is used to modulate the vibration strength (VS) according to the principle as shown in formula

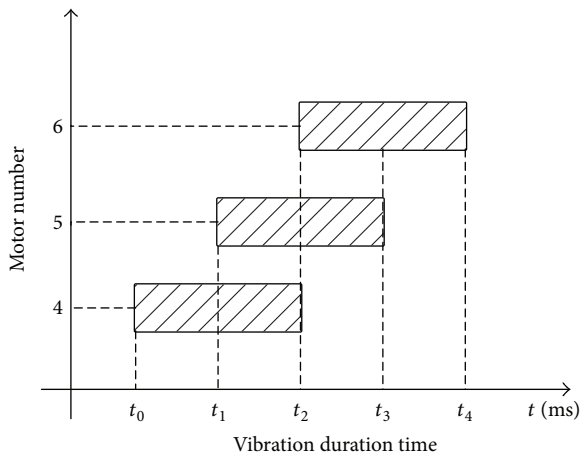


Figure 16: The coding pattern of the sliding tactile representation.

Table 2: Relationship between vibration strength level and voltage applied to motor.

Vibration strength level	Voltage applied to motor (V)
I	1.0
II	2.0
III	3.0
IV	4.0
V	5.0

(19), and the vibration frequency is set to 100 Hz. By this method, the user may feel his/her arm being grasped:

$$\mathrm{VS} = \begin{cases} \mathrm{I} & 0 < r \leq 0.2 \\ \mathrm{II} & 0.2 < r \leq 0.4 \\ \mathrm{III} & 0.4 < r \leq 0.6 \\ \mathrm{IV} & 0.6 < r \leq 0.8 \\ \mathrm{V} & 0.8 < r \leq 1.0, \end{cases} \tag{19}$$

where I, II, III, IV, and V are the vibration strength levels; the relationship between vibration strength levels and the voltage applied to the motor is as shown in Table 2. r is the ratio of grasp force detected by the force sensors to the maximum allowable value of the grasp force:

$$r = \frac{F}{F_{\max}}, \tag{20}$$

where F is the grasp force detected by the force sensors and $F_{\max}$ is the maximum allowable value of the grasp force.

A sliding tactile representation coding pattern is designed according to the phenomenon of apparent movement (see appendix) [19]. The vibration strength is set to level III, and the vibration frequency is set to 100 Hz. As shown in Figure 16, t_0, t_1, and t_2 are, respectively, the beginning time of the motors number 4, number 5, and number 6. The ending times of these three motors, respectively, are t_2, t_3, and t_4. The time intervals of $t_0 \sim t_1$, $t_1 \sim t_2$, $t_2 \sim t_3$, and $t_3 \sim t_4$ are 250 ms. t_0 is the time of sliding situation being detected as well. By this method, the user may feel something sliding on his/her arm.

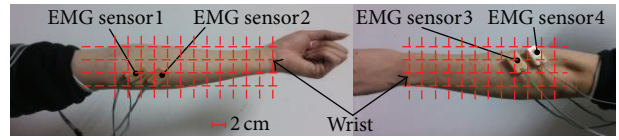


Figure 17: Laminating position of the EMG sensors.

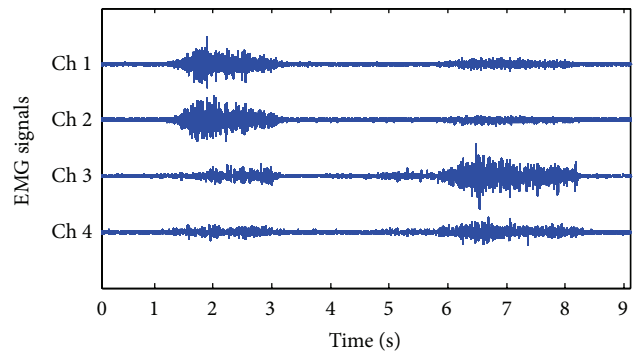


Figure 18: EMG signals coming from 4 EMG sensors.

Table 3: Combination of the EMG signals.

Group	Composition
1	Ch1 and Ch3
2	Ch1 and Ch4
3	Ch2 and Ch3
4	Ch2 and Ch4

4. Experiments and Results

To verify the validity of the control strategy presented in this paper, lab-based experiments were carried out.

4.1. Evaluation Experiments of the EMG Self-Learning Motion Recognition Method. Motion recognition experiment was conducted to verify the effectiveness of the EMG self-learning motion recognition method. In the experiment, four EMG sensors were distributed on the participants' forearms as shown in Figure 17. The participants executed hand motions (hand grasp and hand open) according to the commands. Figure 18 shows the EMG signals when participants executed the hand motions.

According to the laminating positions of the EMG sensors, EMG signals coming from four sensors can be combined into four groups as shown in Table 3. Two recognition methods, including one shown in Figure 1 and the other EMG self-learning recognition method, have been applied to these four groups EMG signals. Figure 19 shows the recognition results by using the recognition method as shown in Figure 1, and Figure 20 shows the recognition results by using the EMG self-learning recognition method.

Table 4: Correlation coefficients between every two recognition results by using the recognition method shown in Figure 1.

Correlation coefficient (%)	Recognition results of group 1	Recognition results of group 2	Recognition results of group 3	Recognition results of group 4
Recognition results of group 1	100	89.39	94.35	87.84
Recognition results of group 2		100	90.49	97.91
Recognition results of group 3			100	91.58
Recognition results of group 4				100

Table 5: Correlation coefficients between every two recognition results by using the EMG self-learning recognition method.

Correlation coefficient (%)	Recognition results of group 1	Recognition results of group 2	Recognition results of group 3	Recognition results of group 4
Recognition results of group 1	100	99.20	99.25	98.11
Recognition results of group 2		100	98.28	99.53
Recognition results of group 3			100	99.04
Recognition results of group 4				100

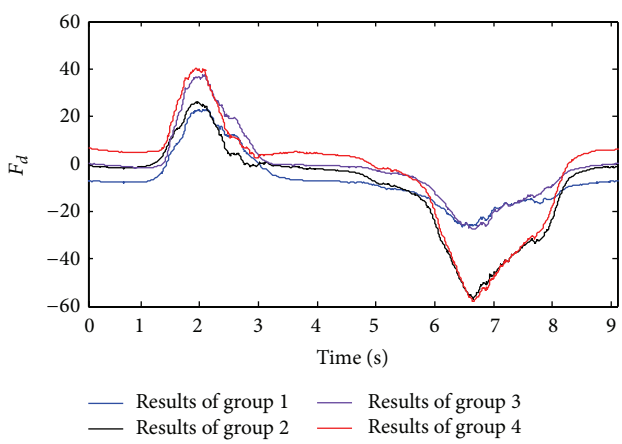


Figure 19: Motion recognition results by using the recognition method shown in Figure 1.

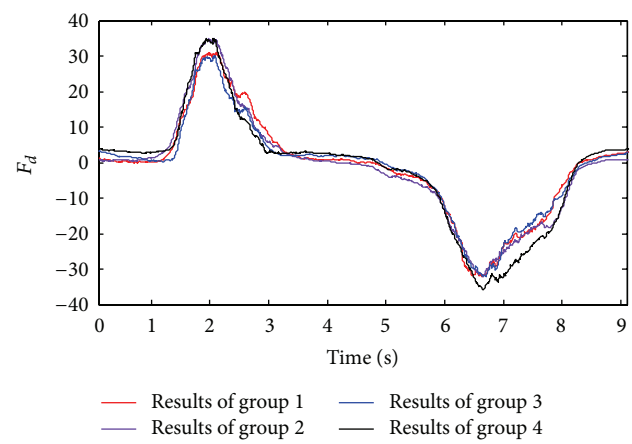


Figure 20: Motion recognition results by using the EMG self-learning recognition method.

To evaluate the consistency of the recognition results, the correlation coefficients between every two recognition results were calculated as shown in Tables 4 and 5. Correlation coefficient (ρ) is a measurement of the linear correlation (dependence) between two variables X and Y. It is widely used as a measurement of the degree of linear dependence between two variables. The formula for ρ is

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y}, \quad (21)$$

where cov is the covariance, σ_X is the standard deviation of X, μ_X is the mean of X, and E is the expectation.

From the results of the EMG motion recognition experiment, recognition results by using the EMG self-learning recognition method have a better consistency. EMG self-learning can reduce the influence on EMG signals caused by the uncertainty of the contacting position of the EMG sensors.

4.2. Evaluation Experiments of the Backstepping Controller. To verify the effectiveness of the BCSFO, velocity tracking and force tracking experiments were implemented. The designed speed was inputted to the controller in free space, while the designed grasp force was inputted to the controller in restricted space. The results are shown in Figures 21 and 22. The objects with different stiffness were grasped by prosthetic hand in the experiment, and the results are shown in Table 6. Table 6 contains the mean value ($\bar{e}$) and mean variance (R) of the force tracking error. The results show that the controller can track the designed velocity and designed grasp force quickly; the tracking error is in the acceptable range.

4.3. Tactile Representation Coding Experiment and Results. Ten nonamputee volunteers (five males and five females, aged from 22 to 27) were chosen to use the MPH-III. The EMG acquisition device was worn on the forearm, and the EMG sensors were put on a pair of antagonistic muscles. The TRD was worn on the upper arm. Five minutes or more was given

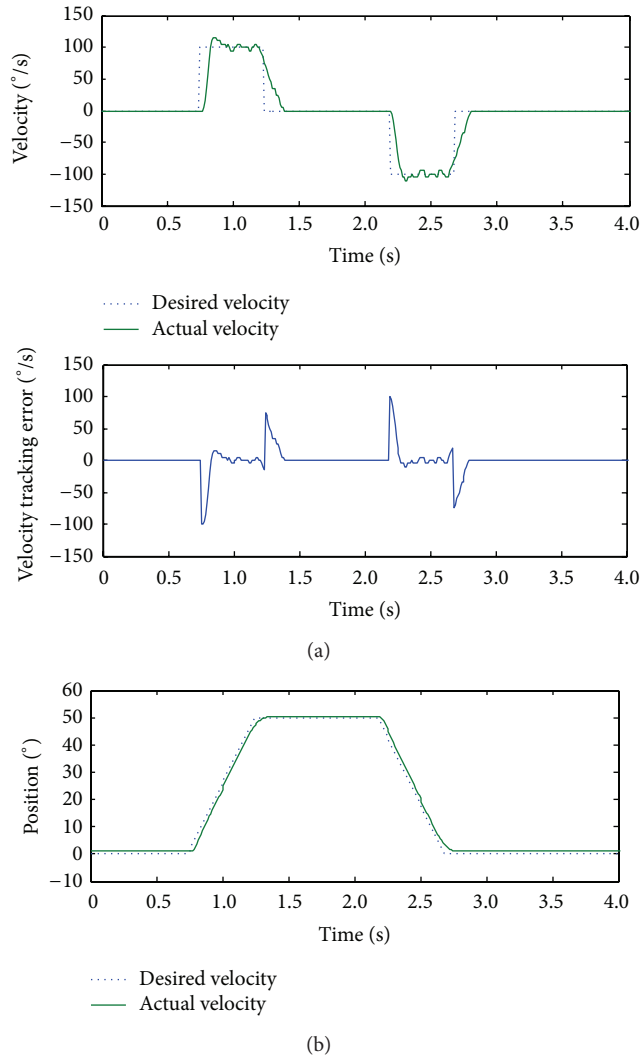


Figure 21: Results of the velocity tracking experiment. (a) Velocity tracking curve and tracking error curve. (b) The expected position and actual position.

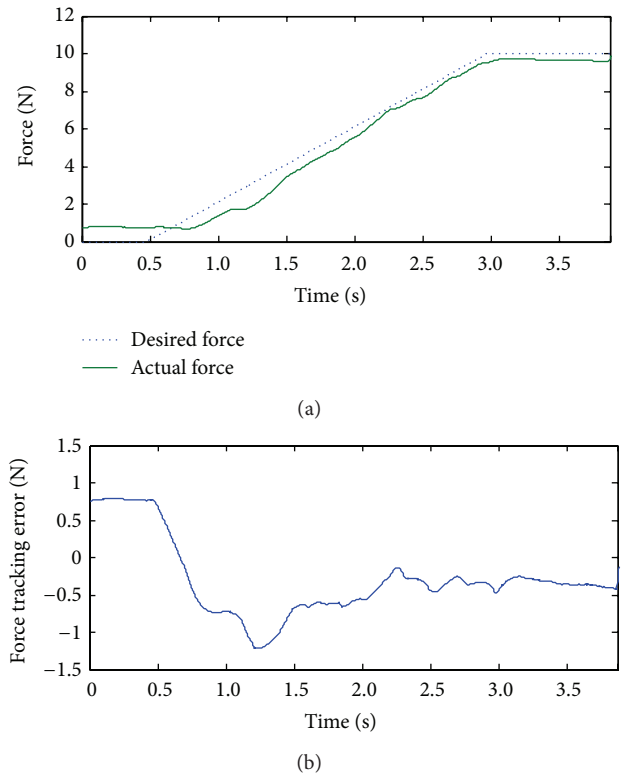


Figure 22: Force tracking curve and tracking error curve.

Table 6: Grasp results of different objects.

Object's number	Object's stiffness (N/mm)	$\bar{e}$ (N)	R (N)
1	0.5	0.82	1.00
2	1	1.46	1.33
3	2	0.88	1.02
4	4	1.43	1.27
5	10	1.22	1.10

to each participant to get familiar with the prosthetic hand. After experiments, they graded these three control strategies.

The TRD outputs the grasp force vibration coding for 25 times. Among them, the sliding state vibration coding occurs for five times randomly. The participants were asked to record the vibration state according to their feelings. Each vibration state occurs 5 times. Each participant accomplished experiments without influence from others. The results of the experiment are shown in Figures 23 and 24.

In Figure 23, x-axis represents the actual strength level outputted by the TRD; y-axis represents the times the participants recorded the strength level. The different strength levels the participant felt are represented by different colors. That is to say, the strength levels I, II, III, IV, and V the participants felt are represented by red, yellow, blue, green, and black, respectively. Red, yellow, blue, green, and black represent the strength levels the participator felt which are I, II, III, IV, and V, respectively.

Take the experimental results of participant 1; for example, when the actual strength level is "I" (see x-axis), the times of the strength levels I, II, III, IV, and V perceived by the participants are 5, 0, 0, 0, and 0, respectively. When the actual strength level is "II," the times of the strength levels I, II, III, IV, and V perceived by the participants are 0, 5, 0, 0, and 0, respectively. When the actual strength level is "III," the times of the strength levels I, II, III, IV, and V perceived by the participants are 0, 0, 4, 1, and 0, respectively. When the actual strength level is "IV," the times of the strength levels I, II, III, IV, and V perceived by the participants are 0, 0, 1, 3, and 1, respectively. When the actual strength level is "V," the times of the strength levels I, II, III, IV, and V perceived by the participants are 0, 0, 0, 1, and 4, respectively.

Figure 24 shows the average values of each strength level perceived by the participants in each actual strength level.

The results of Figures 23 and 24 show that the participants can distinguish the strength level precisely when the strength level of tests is given in low level (levels 1 and 2), while on the condition of high level (levels 3 to 5), the strength

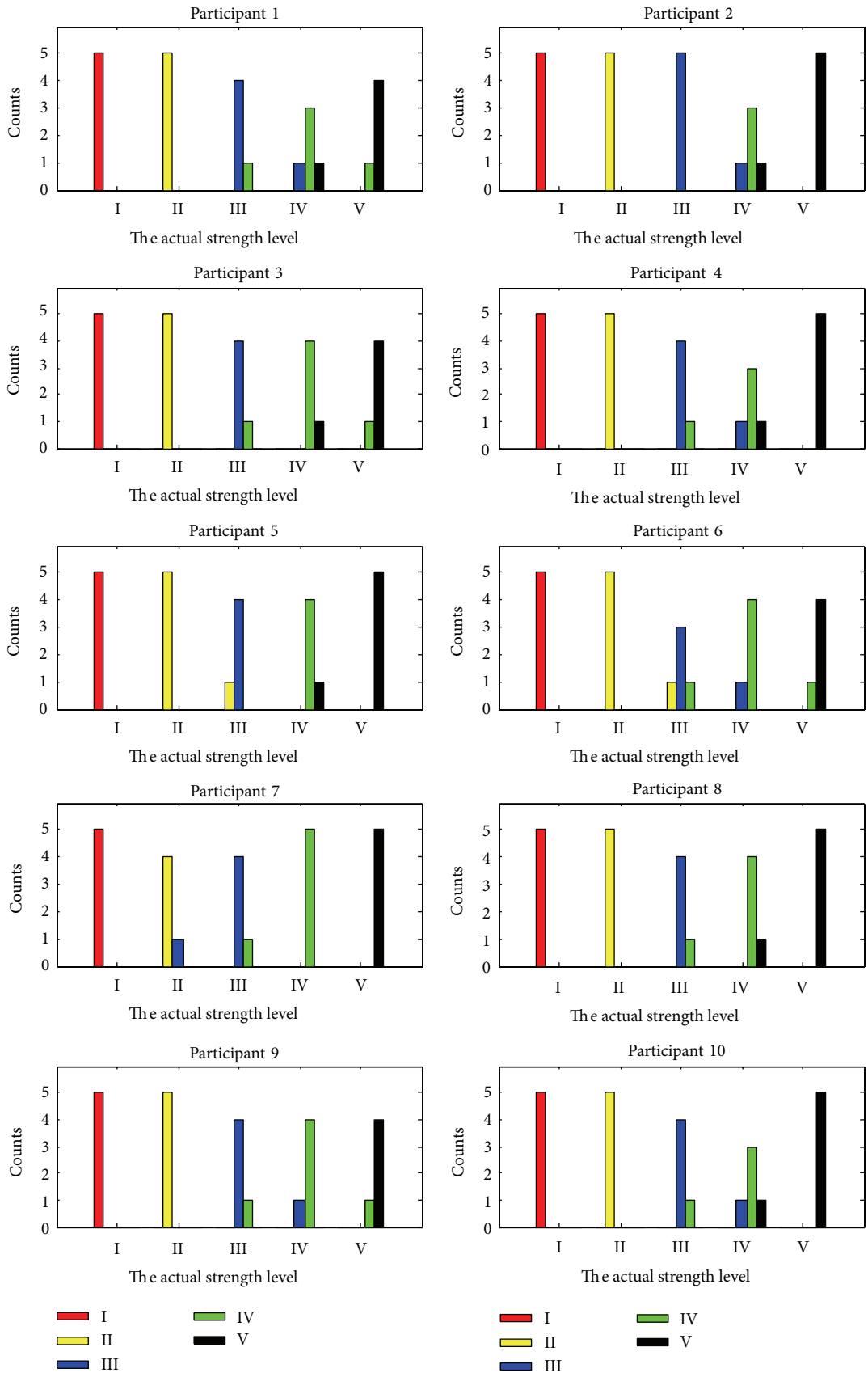


Figure 23: Results of the grasp force coding experiment.

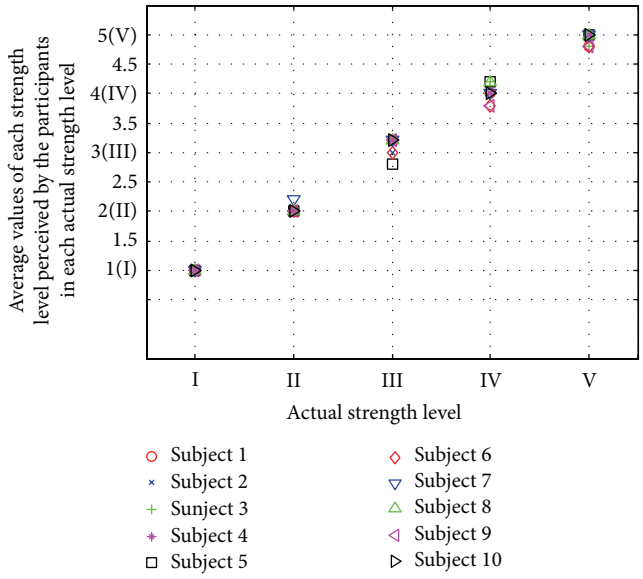


Figure 24: Average values of each strength level perceived by the participants in each actual strength level.

Figure 25: Experiment scene of grasping.

level perceived by the participants in each actual strength level fluctuates around the actual strength level. The errors between false records and correct records are no more than 1 level. The results of all the participants show the same trend. In the view of the experimental results, this kind of vibration coding can help users to perceive the grasp force of the prosthetic hand.

The experimental result of tactile representation of sliding state is as its expected because the coding pattern is simple and there is no relationship between the siding coding pattern and grasp force tactile feedback. Most of the participants can make the judgment whether the sliding occurs or not.

4.4. Grasp Experiments and Results. The force control strategy (M1) shown in Figure 1, the EMG self-learning control strategy (M2), and control strategy shown in Figure 2 (M3) were compared with each other. These three strategies all contain visual feedback. The differences are that M2 has EMG self-learning function and the controller in M2 is BCSFO. Besides the features of M2, M3 has an additional function of tactile feedback. The experiments assume that the functions of visual feedback in these three strategies are the same. Five minutes was given for participants to get familiar with

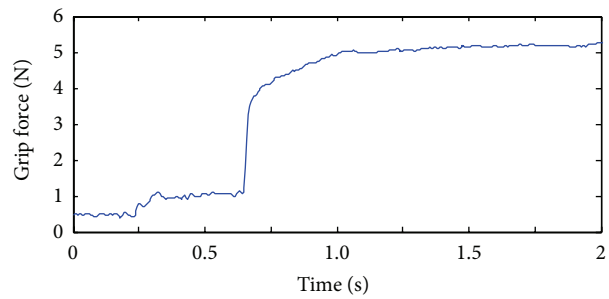


Figure 26: Grasp force curve.

these control strategies. Then, ten participants were asked to grasp and lift a paper cup which was full of water (see Figure 25). In this process, the participant should keep water without overflowing or dropping. All control strategies were tested for ten times. The number of successful lifting was used to quantify the performance of the control strategy. And the participants were asked to evaluate every control strategy after experiments according to their feeling about the performance of the control strategy. And the performance includes flexibility and usability. The best control strategy was marked as highest grade, 10. Table 7 shows the results of the experiment. Figure 26 shows the grasp force curve which was recorded in the process of a participant grasping a paper cup by using the control strategy of M3.

In this experiment, the comments from all participants show that the M3 is the best one among these three strategies. When using prosthetic hand without EMG self-learning, most of the participants have to adjust the positions of the EMG sensors on the skin surface, while it is unnecessary to adjust by using the strategies with EMG self-learning.

5. Conclusions

The control strategy with tactile feedback for EMG prosthetic hand is described in detail. Aiming at reducing the influence on the EMG signals which comes from the attaching positions of EMG sensors, an EMG self-learning recognition method is proposed. A BCSFO is proposed to realize the velocity proportional control in free space and grasp force tracking control in restricted space. A tactile representation system is designed to help the user perceive the state of the prosthetic hand, and the states include grasp force and slid information.

The experiments are implemented to verify the effect of the proposed control strategy. And the results show that the different contacting positions between sensors and arm lead to the variance of the EMG signals, and this kind of influence can be reduced by the proposed EMG self-learning method. The proposed BCSFO can meet the requirements of the prosthetic hand (velocity proportional control in free space and grasp force tracking control in restricted space). And the results of the grasping experiments show that the strategy with EMG self-learning method and tactile feedback (M3) is better than the strategy of the force control (M1) and

Table 7: Results of the paper cup grasping experiment.

	Mean success rate of paper cup grasping (%)	Standard deviation of paper cup grasping (%)	Mean score marked by the test participants
Force control (M1)	51	18	6.35
EMG self-learning force control (M2)	59	20	7.25
EMG self-learning force control with tactile feedback (M3)	79	15	10

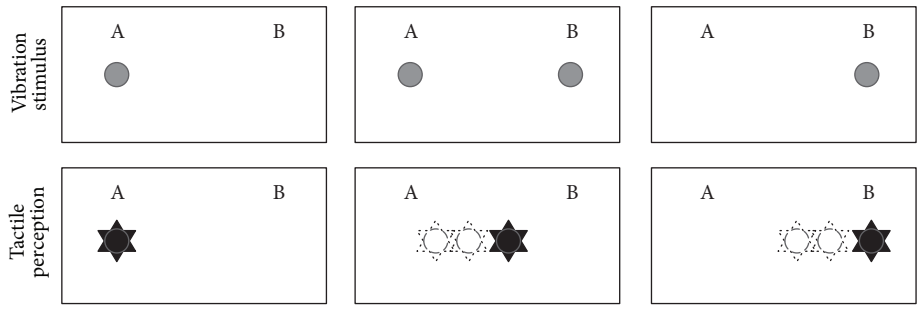


Figure 27: Diagram of the apparent movement.

the strategy of the EMG self-learning control (M2) in the aspect of control effect of the prosthetic hand.

Moreover, all the participants think that the EMG signal self-learning pattern recognition method is much more helpful and convenient in the process of manipulating the prosthetic hand.

For the future work, after lots of experiments we will research a more effective coding pattern for tactile representation, which would be easily accepted by amputees.

Appendix

This phenomenon was described in detail in [19]. Its basic principle is as follows.

(1) Point A starts to vibrate.

(2) Point B starts to vibrate.

(3) Point A stops.

By applying vibration stimulation in this order, the participant will get an illusion that point A is moving towards point B (see Figure 27).

Conflict of Interests

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

Acknowledgment

This paper was supported by the Nature Science foundation of China under Grant no. 61325018.

References

[1] H. H. Sears and J. Shaperman, "Proportional myoelectric hand control: an evaluation," *American Journal of Physical Medicine and Rehabilitation*, vol. 70, no. 1, pp. 20–28, 1991.

[2] M. H. M. Zaini and S. A. Ahmad, "Surgical and non-surgical prosthetic hands control: a review," in *Proceedings of the IEEE Symposium on Industrial Electronics and Applications (ISIEA '11)*, pp. 634–637, 2011.

[3] D. Graupe and W. K. Cline, "Functional separation of EMG signals via ARMA identification methods for prosthesis control purposes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 5, no. 2, pp. 252–259, 1975.

[4] T. Khoshaba, K. Badie, and R. M. Hashemi, "EMG pattern classification based on back propagation neural network for prosthesis control," in *Proceedings of the 12th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1474–1475, November 1990.

[5] K. Mahaphonchaikul, D. Sueaseenak, C. Pintavirooj, M. Sangworasil, and S. Tungjitkusolmun, "EMG signal feature extraction based on Wavelet transform," in *Proceedings of the 7th Annual International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '10)*, pp. 327–331, May 2010.

[6] A. Fassih, D. S. Naidu, S. Chiu, and M. P. Schoen, "Precision grasping of a prosthetic hand based on virtual spring damper hypothesis," in *Proceedings of the 5th Cairo International Biomedical Engineering Conference (CIBEC '10)*, pp. 79–82, Cairo, Egypt, December 2010.

[7] C.-H. Chen, D. S. Naidu, A. Perez-Gracia, and M. P. Schoen, "A hybrid adaptive control strategy for a smart prosthetic hand," in *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '09)*, pp. 5056–5059, IEEE, September 2009.

[8] E. D. Engeberg and S. G. Meek, "Backstepping and sliding mode control hybridized for a prosthetic hand," *IEEE Transactions on*

Neural Systems and Rehabilitation Engineering, vol. 17, no. 1, pp. 70–79, 2009.

[9] C. Wu, A. Song, and H. Zhang, "Adaptive fuzzy control method for EMG prosthetic hand," *Chinese Journal of Scientific Instrument*, vol. 34, no. 6, pp. 1339–1345, 2013.

[10] J. Liu and A. Song, "Discrimination and memory experiments on haptic perception of softness," *Perceptual and Motor Skills*, vol. 106, no. 1, pp. 295–306, 2008.

[11] C. Antfolk, M. D'alonzo, B. Rosén, G. Lundborg, F. Sebelius, and C. Cipriani, "Sensory feedback in upper limb prosthetics," *Expert Review of Medical Devices*, vol. 10, no. 1, pp. 45–54, 2013.

[12] A. Song, J. Wu, G. Qin, and W. Huang, "A novel self-decoupled four degree-of-freedom wrist force/torque sensor," *Measurement*, vol. 40, no. 9-10, pp. 883–891, 2007.

[13] J. Ma, A. Song, and D. Pan, "Dynamic compensation for two-axis robot wrist force sensors," *Journal of Sensors*, vol. 2013, Article ID 357396, 5 pages, 2013.

[14] A. Song, Y. Han, H. Hu, and J. Li, "A novel texture sensor for fabric texture measurement and classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, pp. 1739–1747, 2013.

[15] H. Hu, Y. Han, A. Song, S. Chen, C. Wang, and Z. Wang, "A finger-shaped tactile sensor for fabric surfaces evaluation by 2-dimensional active sliding touch," *Sensors*, vol. 14, no. 3, pp. 1588–1592, 2014.

[16] D. Wood, "Tactile displays: present and future," *Displays*, vol. 18, pp. 125–128, 1998.

[17] S. A. Brewster and L. M. Brown, "Tactons: structured tactile messages for non-visual information display," in *Proceedings of the 5th Australasian User Interface Conference (AUIC '04)*, pp. 15–23, Dunedin, New Zealand, 2004.

[18] K. MacLean and M. Enriquez, "Perceptual design of haptic icons," in *Proceedings of Eurohaptics*, pp. 351–363, 2003.

[19] G. V. Bekesy, "Sensation on the skin similar to directional hearing, beats, and harmonics of the ear," *Journal of the Acoustic Society of America*, vol. 29, no. 4, pp. 489–501, 1957.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 846487, 12 pages
http://dx.doi.org/10.1155/2015/846487

Review Article

Pressure Sensor: State of the Art, Design, and Application for Robotic Hand

Ahmed M. Almassri,[1] W. Z. Wan Hasan,[1] S. A. Ahmad,[1] A. J. Ishak,[1] A. M. Ghazali,[1] D. N. Talib,[1] and Chikamune Wada[2]

[1] *Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia*
[2] *Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan*

Correspondence should be addressed to Ahmed M. Almassri; eng.ahmed8989@gmail.com and W. Z. Wan Hasan; wanzuha@upm.edu.my

Received 5 November 2014; Accepted 15 December 2014

Academic Editor: Guangming Song

We survey the state of the art in a variety of force sensors for designing and application of robotic hand. Most of the force sensors are examined based on tactile sensing. For a decade, many papers have widely discussed various sensor technologies and transducer methods which are based on microelectromechanical system (MEMS) and silicon used for improving the accuracy and performance measurement of tactile sensing capabilities especially for robotic hand applications. We found that transducers and materials such as piezoresistive and polymer, respectively, are used in order to improve the sensing sensitivity for grasping mechanisms in future. This predicted growth in such applications will explode into high risk tasks which requires very precise purposes. It shows considerable potential and significant levels of research attention.

1. Introduction

Robotic hand is a mechatronic machine that is made to complete assignment whenever it is required, especially for repetitive and dangerous tasks, and also during specific applications such as military robots, home automation [1], automotive industries, and nuclear industry robots. In fact, many robotic hand applications were already developed as in [2, 3]; for instance, dexterous manipulation [4–6], tactile image perception [7, 8], artificial limbs [9], fingerprint recognition [10], grasping objects [11–13], and pick and place applications [14, 15] can also be widely seen in various industries. Nonetheless, some of these robotic applications require a lot of labor force, notably in terms of assembly line and material handling.

Henceforth, there is a significant need to have a dedicated machine, which is suitable for robust robotic application. For example, robotic hands manufacturing industries for pick and place mechanism are programed to complete task where it takes a product from one spot and put it to a different location. This technology has the advantages of reducing the risk process associated with human operators during the manufacturing process. Besides, it also saves time and energy required for the labor. Therefore, tactile sensing in the robotic hand is defined as a sensor device that is good enough to measure various properties of an object and provide information through physical touch between a sensor and an object [16]. Recently, the enhancement of the robotic hand sensor has received a substantial attention and becomes crucial to our everyday life. Researchers have recognized that equipping a robot with different sensors is a way to perform tasks in unstructured environment and enable the robot to cope with significant uncertainties. Due to the demand of ensuring safety between robots and objects during the mechanical touch, intelligent tactile sensing in robotic hand with high capabilities is critical.

In this paper, the different techniques for measuring force or interface pressures are presented. These techniques include

load cells, pressure indicating film, and tactile pressure system. Similarly, a review on industry pressure sensing that involves the pick and place applications and algorithm control is also highlighted. The paper also discusses the MEMS sensor technology and different types of sensors while the last section of this part discusses the piezoresistive flexiForce sensor. FlexiForce sensor has a good substrate material, which is a polymer that enhances the force sensing and improves the performance of force, linearity, hysteresis, drift, and temperature sensitivity compared to any other thin film. Furthermore, it is flexible and ultrathin enough as the researchers and designers can use it in different integrated applications as well as for applications that are oriented to manipulative tasks with grippers of robotic hand. In a nutshell, new applications for tactile pressure sensing show a high increase in publications and research attention as viewed in Table 2. As a result, the design of sensor becomes more precise with higher reliability to overcome the problems.

2. Sense of Pressure: Methods

Pressure is force per unit area applied in a direction perpendicular to the surface of an object. The formula is commonly written as follows:

$$P = \frac{F}{A}, \tag{1}$$

where P is the pressure, F is the normal force, and A is the area of the surface of contact. When two objects are contacted, they exert force on each other. Thus, the average interface pressure is the total force divided by the interface region. In contrast, pressure measurement is necessary to get a peak pressure when the interface pressure is not uniformly distributed. In this context, there are three technologies and methods to be considered to measure force or interface pressures, load cells, pressure indicating film, and tactile pressure sensor.

2.1. Load Cell. Load cell is a type of pressure sensor, which is commonly used in industrial weighing product to measure force such as goods and vehicles. The gripper of a robotic hand that picks up an object can be equipped with load cells in order to provide compression force feedback to the control system which prevents damage to the object or released too early. Also, load cells can be used to measure the compression forces during a robot walk to provide data for the equilibrium-controlling system. In industrial machinery, rods, beams, wheels, and bars are instrumented in order to control the forces exerted on them. Due to this variety of possible applications, load cells are very important [23]. There are many types of technologies which are used to measure loads such as strain gauges, piezoelectric elements, and variable capacitance.

Moreover, depending on the applied force and mechanics of application, multiple form factors of load cells are utilized. Typically, multiaxis MEMS force-torque sensors are used to measure the load. In the literature, a small number of multiaxis MEMS sensors have been reported. In [24], a capacitive MEMS force sensor is presented, which assess the force along two axes. In [25, 26], the design of a piezoresistive three-axis force sensor is described. A piezoresistive torque sensor has been presented in [27]. In addition, a three-axis capacitive MEMS force-torque sensor has been reported in [28] and this sensor is able to measure forces along two axes and a torque perpendicular to these forces. As we know, forces can be sensed in a plane, while a torque perpendicular to this plane can be measured. In certain conditions, researchers can use multiple load cells to measure forces over multiple regions of a contact interface [29]. In Figure 1, an interface or contact area is divided into four quadrants with an exclusive load cell measuring each area. This arrangement provides more details on the force distribution across the surface; the average pressure in each zone will occur. Still, the disadvantage of this technique is inconsistent because the load cell can show the total force but cannot identify localized spikes in pressure. There are many different types of load cells that operate in different ways, but currently the most commonly used load cell is the strain gage (or strain gauge) load cell.

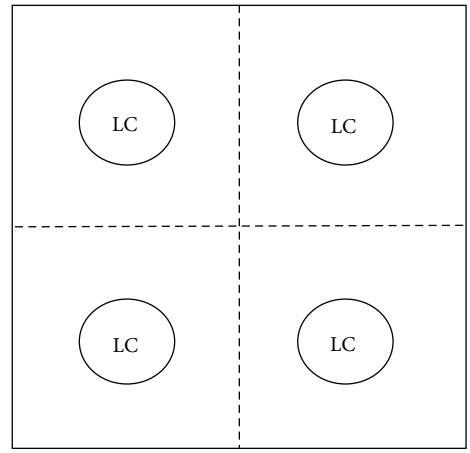


Figure 1: Pressure measurement using multiple load cells [30].

2.1.1. Strain Gauge Load Cells. Strain gages are small patches of silicone or metal that measure mechanical strain and convert the load acting to electrical signals. This load cell is considered as an analog type tool and utilized to measure weight. When a load is applied to a stationary object, stress and strain are the result. Stress is defined as the object's internal resisting forces, and strain is the displacement and disfigurement that occurs [31], so the load causes deformations in the material or object that can be measured using strain gauges. Two capacitive pressure gauges which have been used extensively in the study of liquid and solid helium are described [32]. Figure 2 illustrates a structure of strain gauges. Here, more resistance in strain gauge will increase the stability as seen in Figure 2. In fact, it also offers a wide range of different patterns which means various applications will occur.

2.2. Pressure Indicating Film. Pressure indicating film is widely used to measure interface pressures between two

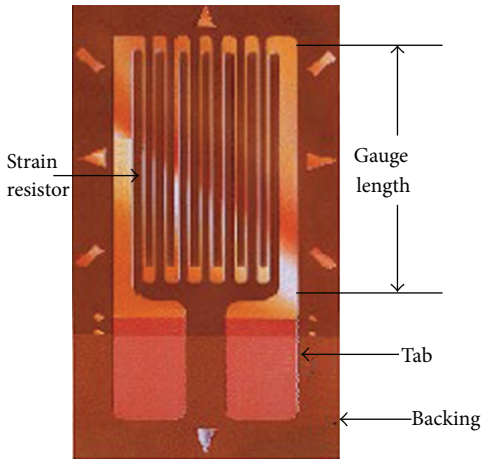


Figure 2: Structure of strain gauge [31].

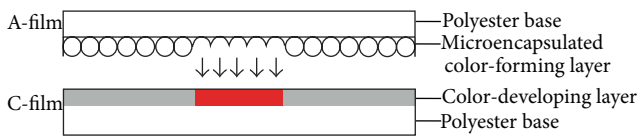


Figure 3: Components of pressure indicating film [30].

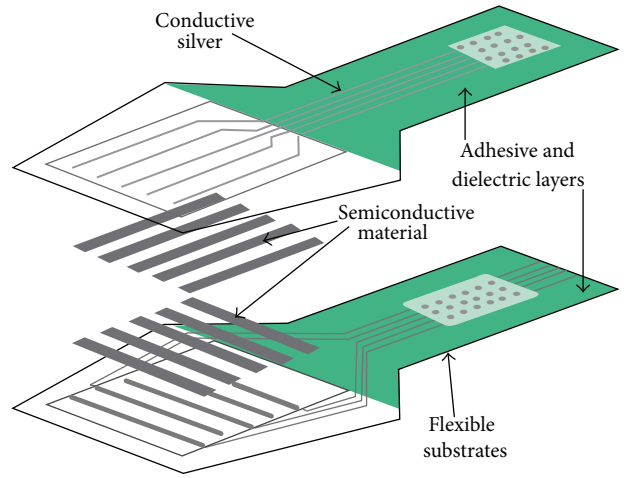


Figure 4: Construction of a tactile pressure sensor [30].

surfaces. Two sheets of polyester are designed to measure the force applied across the sensing area. Figure 3 shows a pressure indicating film. Here, a colour material, under a layer of polyester, is layered next to tiny microcapsules, which are utilized to break under different pressures. When pressure is applied to the film, the microcapsules are broken and distributed ink where pressure is deformed, and the colour intensity of the resulting image reveals the relative amount of applied pressure. As a result, the greater the pressure, the darker the colour and an image of the force applied will be composed across the sensing area. Various features cause the pressure indicating film to be used in a wide range of applications including flexibility, being easy to use, and thinness that plays a major role in capturing image of applied pressure. Furthermore, there are no attached electronics, that is, a good material or film to obtain the pressure distribution without concern of crushing wires or expensive electronics during the film feeding through rollers. Pressure indicating film is used in applications that requires static pressure measurements, visual pattern of peak pressure, and one-time use [30].

2.3. Tactile Pressure Sensor. Tactile pressure sensor measures various parameters of an object through physical touch between sensor and an object [16]. The measured parameters are, namely, pressure, temperature, normal and shear forces, vibrations, slip, and torques. In this context, pressure and torque are example of an important parameter, and it is typically measured through physical touch. Detection and measurement of a point contact force can also be considered as a part of touch sensor for pressure and torque, but then again, tactile sensing can also ease up the process of interpretation of the corresponding information for the parameter.

By definition, tactile sensing means an array of a coordinated group of touch sensors [33]. A common type of tactile pressure sensor consists of an array sensor. For instance, Figure 4 shows a unique piezoresistive material sandwiched between two pieces of flexible polyester; each half of them has printed silver conductors. The result is a very thin 0.004″ (0.1 mm) sensor, which can be used in various applications, especially for industrial and medical robot. A conductive track, which is composed of silver conductors, will scan the electronics and transmit a signal through the piezoresistive ink. The tactile array sensor signal can be processed to offer a great deal of parameters about contact kinematics and precise tactile information for robotics, haptic feedback, and other contact applications. Among the parameters that can be extracted are contact location, object shape, and effective width of the pressure distribution.

The pressure distribution can be achieved by identifying the position of all the applied forces. To do this, an array sensor which has vertical and horizontal of piezoresistive traces is needed. Here, each row and column intersecting in one point are called a sensel. More intersection implies more sensels which means the more spatial resolution is the sensor. Because of many human machine interfaces (e.g., wheelchair seating systems, driver's seats, bed mattresses, hospital beds) [34, 35] and because human joint is incongruent, the sensor should be in a wide range of resolutions. However, the tactile pressure sensor array has a good spatial resolution of the pressure distribution. Figure 5 illustrates the sensing system and an electrical schematic of electronics that scan each sensel. When force is applied to the sensel, the sensel which is represented by a variable resistor will be changed and the possible current will flow through the device, and then the electronics will collect the analog data, which can be compensated with proper calibration.

From the reviews that have been obtained, there are several factors to be considered, specifically on technology for interface force and pressure measurement between two

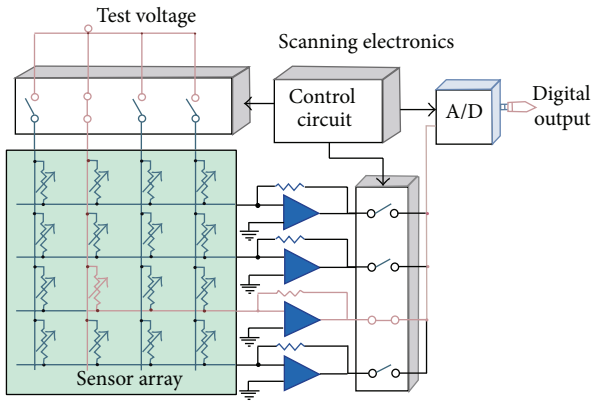


Figure 5: Electric schematic of sensor [30].

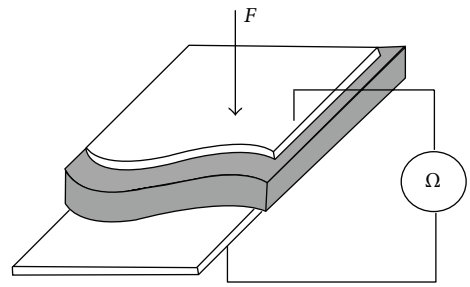


Figure 6: Piezoresistive tactile pressure sensor [38].

surfaces for robotic hand applications purposes. Comparisonwise, load cells provide the most reliable data pressure measurement, but the size and number of load cells limit the density of measurement points. The total load can be easily reported; however, the size of the load cell can be a limiting factor when it reaches fine granularity due to its pressure distribution.

Pressure indicating film can be used in variety of applications such as robotic hand, but the nature of the film will only provide the peak pressure between interfaces during a measurement. This has obvious limitations when trying to measure dynamic applications and also the resulting data pressure measurement has less accuracy, whereas tactile pressure sensor can provide detailed dynamic measurements of interface pressure with minimal impact on system dynamics. The sensing elements need to be properly calibrated to provide accurate data, but the resulting measurements will provide the most in-depth analysis of interface system dynamics. Depending on the information needed and the physical constraints of the system being measured, load cells, pressure indicating film, and tactile pressure sensors each have advantages and constraints for providing accurate and meaningful data pressure measurement. Understanding how these strengths and limitations influence an application is crucial.

3. Pressure Sensor: Design and Technology

A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or an instrument. There are various types of sensors: thermal sensor, electromagnetic sensor, mechanical sensor, pressure sensor, and others. Pressure is sensed by mechanical elements such as plates, shells, and tubes that are designed and constructed to deflect when pressure is applied. Deflection of the elements must be transduced to obtain an electrical or other output. Pressure sensor can differ in technology, design, performance, application suitability, and cost. It can be classified based on various transduction principles such as resistive/piezoresistive, tunnel effect, capacitive, optical, ultrasonic, magnetic, and piezoelectric. The relative merits and demerits of different transduction methods are given in Table 1. Worldwide, there are hundreds of different technologies used in pressure sensor designs such as sensing element method, material, MEMS, nanotechnology, and others. Furthermore, there are significant differences in the types of pressure sensor results from different material used as well as their functional properties.

Recently, pressure sensor using MEMS technology has received enormous attention due to various advantages over traditional electromechanical sensing technology. MEMS offers small size, low weight, low cost, high performance, large scale integration, low power consumption, wider operating temperature, and higher output signal [36]. This section discusses in detail the three main types of pressure sensor methods for robotic hand application: piezoresistive, piezoelectric, and capacitive pressure sensor. Also, various designs and technologies of the pressure sensor are explained in this section.

3.1. Piezoresistive FlexiForce Sensor. As sensor technology grows these days, there are many types of sensors that have been used for pick and place application, especially resistive method due to its stability and high sensitivity. Piezoresistive sensors use the change of the electrical resistance in a material when it has been mechanically deformed. The resistance of a piezoresistor is given as follows:

$$R = \frac{\rho \times l}{t \times \omega}, \tag{2}$$

where ρ, l, t, and ω denote the resistivity, length, thickness of the piezoresistor, and the width of the contact, respectively. Figure 6 shows an example of the piezoresistive tactile pressure sensor. Due to the various features of piezoresistive including low cost, good sensitivity, relatively simple construction, long-term stability with low noise, accuracy, and reliability, it shows the maturity of the technology. In addition, the sensor is considered easy to fabricate and integrate with electronic circuit according to the characteristic of the piezoresistive material. However, it can measure only one contact location and it will still need external power. Though this limitation has been improved by [37], that allows measuring many contact points by using parallel analog resistive sensing strips.

The force sensing resistor (FSR) is based on piezoresistive sensing technology. It can be made in a variety of shapes and

Table 1: Relative merits and demerits of various tactile sensor types [17].

Type	Merits	Demerits
Resistive	(i) Sensitive (ii) Low cost	(i) High power consumption (ii) Generally detect single contact point (iii) Lack of contact force measurement
Piezoresistive	(i) Low cost (ii) Good sensitivity (iii) Low noise (iv) Simple electronics	(i) Stiff and frail (ii) Nonlinear response (iii) Hysteresis (iv) Temperature sensitive
Tunnel effect	(i) Sensitive (ii) Physically flexible	(i) Nonlinear response
Capacitive	(i) Sensitive (ii) Low cost (iii) Availability of commercial A/D chips	(i) Hysteresis (ii) Complex electronics
Optical	(i) Physically flexible (ii) Sensitive (iii) Fast (iv) No interconnections	(i) Loss of light by micro bending chirping (ii) Power consumption (iii) Complex computations
Ultrasonic	(i) Fast dynamic response (ii) Good force resolution	(i) Limited utility at low frequency (ii) Complex electronics (iii) Temperature sensitive
Magnetic	(i) High sensitivity (ii) Good dynamic range (iii) No mechanical hysteresis (iv) Physical robustness	(i) Suffer from magnetic interference (ii) Complex computations (iii) Somewhat bulky (iv) Power consumption
Piezoelectric	(i) Dynamic response (ii) High bandwidth	(i) Temperature sensitive (ii) Not so robust electrical connection
Conductive rubber	(i) Physically flexible	(i) Mechanical hysteresis (ii) Nonlinear response

sizes and can be utilized in many applications in order to measure a proportional change in force and rate of change and also detects contact or touch between objects. FlexiForce manufactured Tekscan is one of the most piezoresistive sensors widely used in robotic hand. Figure 7 shows a tactile force sensor or FlexiForce sensors. This sensor is considered one of the best ideal force sensors for designers, researchers, or anyone who needs to measure forces. With its thin construction, flexibility, and force measurement ability, the FlexiForce sensor can measure the force between any two surfaces and is resilient to most environments. FlexiForce has better force sensing properties, linearity, low hysteresis, drift, and temperature sensitivity than any other thin film force sensors according to the good substrate material which is a polymer. This material has been considered suitable enough to use in robotic hand for grasping objects effectively.

The structure of the force sensor is a substitute of a matrix of sensing traces; the ink uniformly covers an area to measure the total force applied to that space. The sensor consists of two layers of substrate as shown in Figure 7. This substrate is formed of polyester film and a conductive material, silver, which applies to each layer. Layer of pressure sensitive ink is then used, followed by adhesive to combine the two layers of substrate together to compose the sensor. Additionally, the FlexiForce sensor decreases the resistance of the sensing element when the force applied increases. In this context, various applications using FlexiForce sensor are implemented by many researchers [39]. As an example, measurement of interface pressure or force between two soft objects is presented in [40]. Teleoperated robotic systems using tactile force sensor for the design and development of a low cost control rig to intuitively manipulate an anthropomorphic robotic arm with gripping force sensing are reported in [41]. The measurement of low interface pressure between the skin support surfaces and pressure garments is also discussed in [42]. Thereupon, one good example of using FlexiForce sensors is pick and place application which offers to achieve high sensitivity and minimize slip movement and weight measurement with a secure grasp.

3.2. Piezoelectric Pressure Sensor. Piezoelectric sensors convert an applied stress or force into an electric voltage [43]. Piezoelectric material is considered a smart material due to its property which can be used as a sensor and actuators. Furthermore, the piezoelectric materials also have high sensitivity with high voltage output when force is applied. The sensitivity of piezoelectric force sensors is measured in terms of C/N, with sensitivity reported up to 130pC/N [39]. Piezoelectric is considered as a passive sensor which offers a high reliability that is useful to be applied in various applications. Yet, it is only suitable for detection of dynamic forces because of the voltage output decreases over time [44], because it is not able to measure a static force due to their large internal resistance [45]. Piezoelectric materials like

Table 2: Pressure sensor with various applications based on different transduction methods.

Year	Author/Ref.	Transducer method	Array number of elements	Material type	Force/pressure sensitivity or resolution [N]	Force/pressure sensitivity or resolution range [N]	Application	Limitations	
								Advantages	Weakness
2013	Asadnia et al. [18]	Piezoelectric	2 × 5	Liquid crystal polymer	3 mms^{-1} (No)	—	Underwater sensing (autonomous)	(i) Detect very low frequency (down to 0.1 Hz) in water (ii) High resolution (3 mms^{-1}) (iii) Self-powered and passively sense underwater objects	(i) Applied underwater
2012	Aqilah et al. [19]	Resistive	4 × 4	Conductive rubber	15 kg/cm^2 (No)	—	Robotic hand	(i) Flexibility and stretchability (ii) Linearity	(i) Need external power (ii) Maximum pressure is low
2010	Choi [4]	Strain gauge	4 × 4	Polymer	206.6 mV (No) 70.1 mV (Sh)	0–0.8 (No), (Sh)	Dexterous manipulation (robotic fingertip)	(i) Measure normal and shear loads simultaneously	(i) Low force capacity (0.6 N)
2009	Noda et al. [20]	Piezoresistive	—	Silicon rubber	0.01% (No) 0.1% (Sh)	0.05–3 (No) 0.05–3 (Sh)	Dexterous manipulation (robotic application)	(i) High sensitive sensor	(i) Complex design (ii) Inflexibility
2008	Lee et al. [9]	Capacitive	8 × 8	Polymer	2.5% mN (x-axis) 2.9% mN (y-axis) 3.0% mN (z-axis)	0–0.01 (No), (Sh)	Artificial robotic limbs	(i) Measure normal and shear force distribution	(i) Nonuniform gap between electrodes
2006	Takao et al. [7]	Piezoresistive	6 × 6	Silicon	0.5–1 V/N (No)	0.021–0.176 (No)	Tactile image sensor (robotic fingertip)	(i) High spatial resolution (0.42 mm) (ii) Large scale sensing array	(i) Complex design
2005	Shan et al. [21]	Piezoresistive	4 × 4	Elastic rubber	228 mV/N (No) 34 mV/N (Sh)	0–2 (No)	Nonplanar surfaces (tactile sensor skin)	(i) Detect three-dimensional force (ii) Good flexibility and taped on nonplanar surface	(i) Large area skin with low density of receptors
2000	Kane et al. [8]	Piezoresistive	64 × 64	Polysilicon	1.59 mV/KPa (No) 0.32 mV/KPa (Sh)	0–35 KPa (No) 0–60 KPa (Sh)	Tactile imaging and perception (robotic application)	(i) High resolution shear and normal force (ii) Stress	(i) Sensor array is high (64 × 64) (ii) Need complex signal condition circuit
1997	Rey et al. [10]	Capacitive	128 × 128	Silicon	0.3 bars (No)	—	Fingerprint recognition (sensor application)	(i) Low cost (ii) Simple construction	(i) Low resolution (0.05 mm) (ii) Measure normal force only
1992	Fiorillo [22]	Ultrasonic	—	Ferroelectric polymer	3 mm (No)	—	Robotic gripper	(i) Operate at higher frequencies	(i) Low resolution (<3 mm) (ii) Absorption of the ultrasonic signal in air

No: normal force; Sh: shear force.

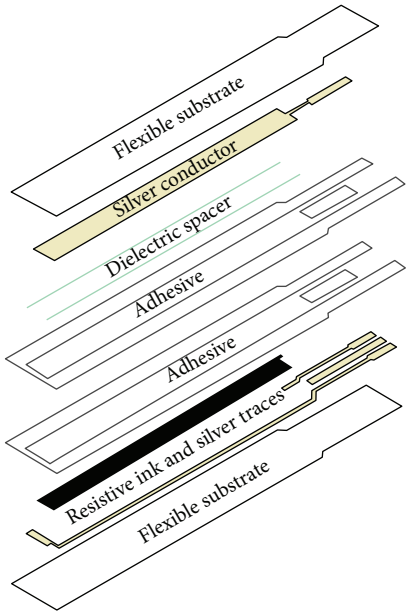


Figure 7: Components of FlexiForce sensor [30].

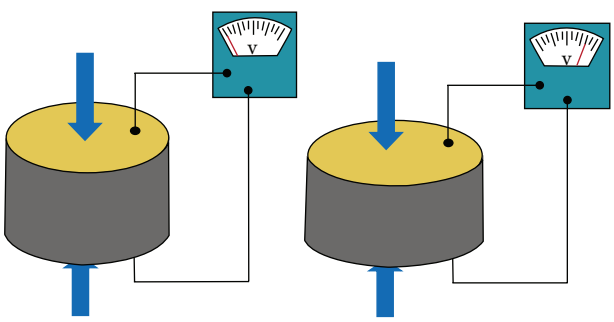


Figure 8: The behaviour of piezoelectric disk based on pressure forces [30].

ceramic lead zirconate titanate (PZT), polymer polyvinylidene fluoride (PVDF), and so forth are suitable for dynamic tactile sensing. Although quartz and ceramic PZT have better piezoelectric properties, the polymer PVDF is more preferred in touch sensors because of their excellent features including mechanical flexibility, high piezoelectric coefficients, dimensional stability, low weight, workability, chemical stability, and chemical inertness [46–48]. The first time PVDF was implemented for tactile sensing technology was reported in [49] and, recently, it was used for environment perception as discussed in [50]. Henceforth, to design the circuit of piezoelectric sensor, an ultrathin input impedance is needed as a considerable effect on response to the device. Figure 8 illustrates that a piezoelectric disk generates a voltage when deformed.

3.3. Capacitive Pressure Sensor. Capacitive sensors consist of a plate capacitor, in which the distance between the plates or electrode area is changed when compressed, and it has a suspended structure that can measure the change in

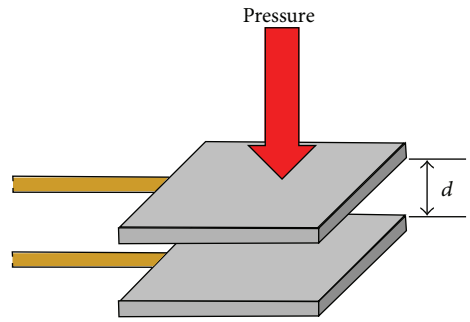


Figure 9: Capacitive sensor with two parallel plates [30].

the capacitance between these two electrodes. For parallel plate capacitors, capacitance can be expressed as follows:

$$C = \frac{\varepsilon o K A}{d}, \tag{3}$$

where C is the capacitance, εo is the relative permittivity of free space constant, K is the dielectric constant of the material in the gap, A is the area of the plates, and d is the distance between the plates. Capacitive tactile pressure sensing is considered as one of the most sensitive techniques for detecting small deflections of structures [51]. It has been developed by researchers for many years due to its features having high spatial resolution, good frequency response, low power consumption, and a large dynamic range [45]. A capacitor sensor array is introduced in [52] and fabricated directly on flexible thin films of polyamide with thickness 25 μm. The sensors show a linear response to applied pressure. Also, few instances of capacitive touch sensors are presented in [53]. Subsequently, an 8×8 capacitive tactile sensing array with spatial resolution at least 10 times better than the human limit of 1 mm is reported in [54]. Two electrodes with air gap, d, are shown in Figure 9.

As summarised, there is no ideal pressure sensor technology that can be used in all applications, since each has specific advantages and constraints. As a matter of fact, the pressure sensor design is primarily determined by the application requirements. It is not just the pressure sensor technology that is vital, but also the practicalities of its implementation of the pressure sensor design must be considered. Furthermore, a wide variety of materials and technologies has been used for the pressure sensor, resulting in performance versus cost tradeoffs. The electrical output signal also provides a variety of choices for various applications.

4. Robotic Hand: Applications

As we know, the human hand is one of the most important parts in a body as it can arrive at narrow places and can execute complex operations. Hence, it is essential for us to have a robotic hand which can accomplish the same procedure as a human hand does in real time. Back in the days, the abilities of humanoid robots focused on walking. Only just, some robotic hands have been developed and there has been a mounting interest in supplying them with

Figure 10: Robotic hand with tactile sensor [33].

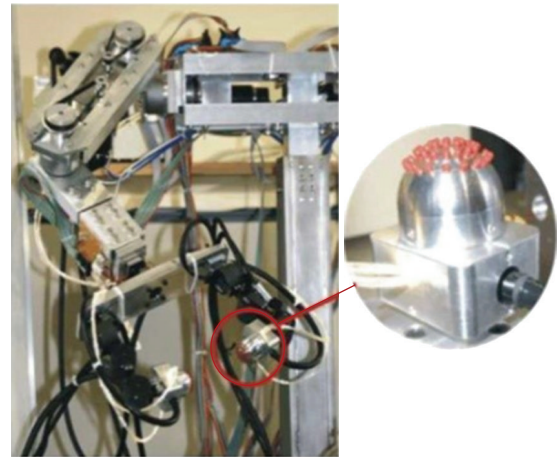


Figure 11: Optical three-axis tactile sensor [64].

proceeding manipulation capabilities [55–58]. Robotic hands have a lot of technologies to execute depending on the required applications, in some instances, picking and sorting cookies [59], military robots [60], welding robots [61], and also nuclear industry robots [62]. Likewise, robotic hands can also be found in various fields such as manufacturing industry, military, space exploration, domestic, transportation, and medical applications.

In this context, manipulation capabilities are one of the robotic hand applications that are central to a robot system. Figure 10 shows an example using commercial products, a robotic hand with tactile sensors from Pressure Profile Systems, Inc. (PPS). This tactile sensing technology gives the robotic hand the ability to manipulate delicate objects without breaking them. Moreover, they will also be able to operate at optimized low powers for energy efficiency by using minimized grasp force. Robotic platform using capacitive sensors is also produced by Pressure Profile Systems, Inc., and it is described in [63].

Next, optical three-axis tactile sensors are shown in Figure 11. It is used to improve sensitization quality in robotic hand system [64]. The arm actuators use the tactile information as feedback to execute and request the orders as needed.

4.1. Tactile Transduction Techniques and Applications. Tactile pressure sensor has been mulled over to be one of the suitable pressure sensors that allow humans to execute dexterous manipulation and offer robot manipulators (hands/fingers) with accurate information on the objects to grab, hold, and handle. Dexterous robotic hands have been developed for the purpose of grasping different objects and it is very challenging for many researchers [65, 66]. Over the years, the change from structured to unstructured environments has made the development of different sensors a priority to enable robots to cope with considerable uncertainties. Because of that, sensors that can get back tactile information have been developed in order to prepare robot hands with such a sense [67].

There are many tactile pressure sensors that are based on a variety of principles such as resistive, capacitive, optical, ultrasonic, magnetic, piezoresistive, or piezoelectric sensors. Up to date, a lot of tactile pressure sensors have been developed. Some of these works, classified on the basis of sensitivity, range of pressure, type of force, and resolution, are given in Table 2. From 1992 to 2013 as viewed in Table 2, methods miscellaneous technologies with various types of materials have been used to sense the pressure in a wide range of applications, particularly in robotic hand. During the time, the size of the sensor design becomes small and delicate. In comparison, an array of 64 × 64 elements was used in tactile imaging and perception with piezoresistive transducer in 2000 as shown in Table 2. On the other hand, after decade of time, technology would have the potential to support the development of more intelligent products in order to improve the quality of human life as in 2012. In this year, many researchers involve their products to achieve what human life needed; a good example is a 4 × 4 array small in size and conformable using the same transducer method comparing with previous example but in a different type of material which is conductive rubber. Additionally, tactile pressure sensor can measure both normal and shear forces produced through dexterous manipulation [68, 69]. However, most of existing tactile sensors only detect the normal contact force during handling objects in a robotic hand manipulation. Hence, the measure of shear force is as important as a normal force to emulate the human hand, which can simultaneously sense the direction and the strength of the applied force within sophisticated manipulation. Likewise, measure of the mechanical contact forces allows controlling the grasp force, which is essential for manipulator displacement and slip movements. This slip detection between fingertip and objects is still one of the main issues that researchers dedicated their time to put an optimal solution to. In fact, this matter required analysing and measuring both shear and normal forces. For this reason, shear information is considered of great importance to full grasp force and torque determination, especially when the pressure does not exceed more than 0.1 N [4].

These parameters will be used to predict and determine slipping of the object.

4.2. Pick and Place Applications. Robotic hands are widely used in manufacturing industry. Typically, it is used for pick and place robot (such as packaging and palletizing). Pick and place process requires the use of robotic hands in order to manipulate objects. In this field, robotic hands have to be programmed in a familiar environment to avoid probable conflict between tools and objects. The agricultural robotic hand is also considered to be a good example to ease up a farmer's job in cutting grass [70]. This robot, which contains a manipulator, a visual sensor, a travelling device, and end-effectors, is able to do a number of tasks by changing the end-effectors. Painting robotic is provided with a door handler of the arm for opening and closing a vehicle door before and after painting it [71]. Furthermore, this system provides a robot for automatic painting vehicle bodies and reduces the equipment's cost by performing the operations of painting vehicle bodies by itself. Similarly, a robot laser welding system is a robot system that consists of a servocontrolled, multiaxis mechanical hand, with a laser cutting head mounted to the faceplate of the robotic hand. The cutting head has focused optics for the laser light and an integral height control mechanism. Most systems use a laser generator that conveys the laser light to the robot cutting head through an optical fiber cable. The benefits of laser welding are that it will produce higher productivity, improved flexibility, and quality welds [72, 73]. All of existing applications used a robotic hand, which are equipped by diverse kinds of sensors to carry through the tasks as necessitated. As a consequence, these missions cannot be accomplished as required without algorithm, since it will be as a framework for the mechanism of robotic hand. Likewise, high performance and safe operating will occur within this algorithm.

4.3. Control Algorithm. From recent development, it is traced that manipulation control is important for a robot. Manipulation control requires some kind of feedback which could provide information about the interaction between the gripper and the grasp objects. This feedback information can be used to implement an algorithm control to achieve the function operator of any robotic hand application as required. It has also been reported that multifingered robotic hand executes particular tasks of grasping an object, which needs to control the measuring required forces for successful operation and dexterous gripper. In addition, it can grasp various objects by changing its shape. Nonetheless, in many cases they lack linearity or sensitivity, especially, in terms of masterful gripper [74]. The robotic hand gripper can increase the sensitivity as well as linearity by using an intelligent feedback control which will be doing the mechanism of gripper object effectively. To wrap up everything, various robotic hand applications using the different algorithm control had been discussed based on tactile sensing capabilities to increase accuracy, flexibility, and receptivity. Moreover, in future robotic hand applications, the manipulators will have to be made lighter and move faster with higher accuracy and work independently. For instance, the automation of complex tasks in industrial applications would be highly enhanced if robots could operate at high speed with high accuracy. Nevertheless, the current robot designs are made massive in size in order to increase rigidity; thereupon, these aims cannot be executed. To achieve high speed operation and faster response for robotic hands manipulations, we should reduce the driving torque requirement. For this purpose, many one-arm flexible robot arms have been built in laboratories [75, 76]. The Shadow Dextrous Robot Hand, in Figure 12, is an advanced humanoid robot hand system available for purchase and is regarded as one of the most advanced robot hands in the current world.

Figure 12: The Shadow Dextrous Robot Hand [77].

In a nutshell, various factors play an important role in achieving the appropriate application as necessitated including material type, transduction method, and conditioning circuit. These factors set the limitation of every application as viewed in Table 2. For instance, the polymer (polyimide) utilized in the tactile pressure sensor fabrication is used for dexterous robotic manipulation applications such as grasping objects, due to it is advantages which are flexible as well as robust enough to withstand forces during grasping. Differently, the silicon MEMS technology used in a tactile image and recognition in robotic applications requires less flexibility but needs high spatial resolution and sensitivity. Hence, the major application of polymer can be applied in wide area tactile sensor like artificial skin and nonplanar surface considered to be having lower fabrication cost than silicon. Moreover, silicon MEMS also reduces the number of electronic signal wires which make it suitable for fingertip and image recognition applications that need high resolution and also suitable for flat surface.

Beside this, the best choice of transduction method and conditioning circuit is very important as they set the limit of power consumption, time response, and number of sensors, allowed to be used in an array. Yet, although piezoresistive sensors are commonly sensitive and economic, they still consume a lot of power rather than others. In addition, they are suitable for detecting dynamic force but have a limitation in the robotic hand application due to voltage

output decreases over time and large internal resistance which make them not able to measure a static force.

5. Conclusion

In this paper, different techniques of pressure sensor types have been reviewed including load cell, pressure indicating film, and tactile pressure mapping system. Similarly, various transaction methods, including piezoresistive, capacitive, and piezoelectric are discussed. Tactile pressure sensor based on piezoresistive material for robotic hand application is also presented. Different materials are used to sense the pressure in many applications including conductive rubber, elastic cantilevers, swollen silicon, elastic rubber, polysilicon, ferroelectric, and polymer. Recently, piezoresistive methods were used for robotic designed especially on grasping objects. It was defined that the advantages were due to making use of polymers which are more flexible, linear, and stretchable. Therefore, it was found that the risk process to workers was reduced and increased safety between the robotic hand and the object during the interaction process.

Conflict of Interests

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

Acknowledgments

This research was supported by a Science Fund grant from the Malaysian Ministry of Science, Technology and Innovation and Grant Putra from the Universiti Putra Malaysia.

References

[1] G. Song, H. Wang, J. Zhang, and T. Meng, "Automatic docking system for recharging home surveillance robots," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 428–435, 2011.

[2] L. Zollo, S. Roccella, E. Guglielmelli, M. C. Carrozza, and P. Dario, "Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 4, pp. 418–429, 2007.

[3] A. Rodriguez, M. T. Mason, and S. S. Srinivasa, "Manipulation capabilities with simple hands," in *Experimental Robotics*, vol. 79 of *Springer Tracts in Advanced Robotics*, pp. 285–299, Springer, Berlin, Germany, 2014.

[4] W.-C. Choi, "Polymer micromachined flexible tactile sensor for three-axial loads detection," *Transactions on Electrical and Electronic Materials*, vol. 11, no. 3, pp. 130–133, 2010.

[5] E. L. Lawrence, I. Fassola, S. Dayanidhi, C. Leclercq, and F. J. Valero-Cuevas, "An evaluation of clustering techniques to classify dexterous manipulation of individuals with and without dysfunction," in *Proceedings of the 6th International IEEE/EMBS Conference on Neural Engineering (NER '13)*, pp. 1254–1257, November 2013.

[6] M. Florek-Jasinska, T. Wimbock, and C. Ott, "Humanoid compliant whole arm dexterous manipulation: control design and experiments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '14)*, pp. 1616–1621, Chicago, Ill, USA, September 2014.

[7] H. Takao, K. Sawada, and M. Ishida, "Monolithic silicon smart tactile image sensor with integrated strain sensor array on pneumatically swollen single-diaphragm structure," *IEEE Transactions on Electron Devices*, vol. 53, no. 5, pp. 1250–1259, 2006.

[8] B. J. Kane, M. R. Cutkosky, and G. T. A. Kovacs, "A traction stress sensor array for use in high-resolution robotic tactile imaging," *Journal of Microelectromechanical Systems*, vol. 9, no. 4, pp. 425–434, 2000.

[9] H.-K. Lee, J. Chung, S.-I. Chang, and E. Yoon, "Normal and shear force measurement using a flexible polymer tactile sensor with embedded multiple capacitors," *Journal of Microelectromechanical Systems*, vol. 17, no. 4, pp. 934–942, 2008.

[10] P. Rey, P. Charvet, M. T. Delaye, and S. A. Hassan, "High density capacitive pressure sensor array for fingerprint sensor application," in *Proceedings of the International Conference on Solid-State Sensors and Actuators*, pp. 1453–1456, Chicago, Ill, USA, June 1997.

[11] S. Hirose and Y. Umetani, "The development of soft gripper for the versatile robot hand," *Mechanism and Machine Theory*, vol. 13, no. 3, pp. 351–359, 1978.

[12] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Visual grasp planning for unknown objects using a multifingered robotic hand," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1050–1059, 2013.

[13] F. Suárez-Ruiz, I. Galiana, Y. Tenzer, L. P. Jentoft, R. D. Howe, and M. Ferre, "Grasp mapping between a 3-finger haptic device and a robotic hand," in *Haptics: Neuroscience, Devices, Modeling, and Applications*, Lecture Notes in Computer Science, pp. 275–283, Springer, Berlin, Germany, 2014.

[14] R. H. Cushman and C. L. Hoegermeyer, "Vacuum pick and place robotic hand," Google Patents, 1985.

[15] Y. Zhang, B. K. Chen, X. Liu, and Y. Sun, "Autonomous robotic pick-and-place of microobjects," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 200–207, 2010.

[16] M. H. Lee and H. R. Nicholls, "Review Article Tactile sensing for mechatronics—a state of the art survey," *Mechatronics*, vol. 9, no. 1, pp. 1–31, 1999.

[17] R. S. Dahiya and M. Valle, "Tactile sensing for robotic applications," in *Sensors: Focus on Tactile, Force and Stress Sensors*, pp. 289–304, InTech, 2008.

[18] M. Asadnia, A. G. P. Kottapalli, Z. Shen, J. Miao, and M. Triantafyllou, "Flexible and surface-mountable piezoelectric sensor arrays for underwater sensing in marine vehicles," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3918–3925, 2013.

[19] A. Aqilah, A. Jaffar, S. Bahari, C. Y. Low, and T. Koch, "Resistivity characteristics of single miniature tactile sensing element based on pressure sensitive conductive rubber sheet," in *Proceedings of the IEEE 8th International Colloquium on Signal Processing and Its Applications (CSPA '12)*, pp. 223–227, March 2012.

[20] K. Noda, Y. Hashimoto, Y. Tanaka, and I. Shimoyama, "MEMS on robot applications," in *Proceedings of the International Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS '09)*, pp. 2176–2181, Denver, Colo, USA, June 2009.

[21] J. H. Shan, T. Mei, L. Sun, L. Ni, M. Meng, and J. R. Chu, "The design and fabrication of a flexible three-dimensional force sensor skin," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1818–1823, August 2005.

[22] A. S. Fiorillo, "Design and characterization of a PVDF ultrasonic range sensor," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 39, no. 6, pp. 688–692, 1992.

[23] I. Muller, R. de Brito, C. E. Pereira, and V. Brusamarello, "Load cells in force sensing analysis—theory and a novel application," *IEEE Instrumentation and Measurement Magazine*, vol. 13, no. 1, pp. 15–19, 2010.

[24] Y. Sun, B. J. Nelson, D. P. Potasek, and E. Enikov, "A bulk microfabricated multi-axis capacitive cellular force sensor using transverse comb drives," *Journal of Micromechanics and Microengineering*, vol. 12, no. 6, pp. 832–840, 2002.

[25] J. Brugger, M. Despont, C. Rossel, H. Rothuizen, P. Vettiger, and M. Willemin, "Microfabricated ultrasensitive piezoresistive cantilevers for torque magnetometry," *Sensors and Actuators A: Physical*, vol. 73, no. 3, pp. 235–242, 1999.

[26] W. L. Tin and C. D. Mote Jr., "Development and calibration of a sub-millimeter three-component force sensor," *Sensors and Actuators A: Physical*, vol. 65, no. 1, pp. 89–94, 1998.

[27] S. Bütefisch, S. Büttgenbach, T. Kleine-Besten, and U. Brand, "Micromechanical three-axial tactile force sensor for micromaterial characterisation," *Microsystem Technologies*, vol. 7, no. 4, pp. 171–174, 2001.

[28] F. Beyeler, S. Muntwyler, Z. Nagy, M. Moser, and B. J. Nelson, "A multi-axis MEMS force-torque sensor for measuring the load on a microrobot actuated by magnetic fields," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3803–3808, San Diego, Calif, USA, November 2007.

[29] J. L. Tan, J. Tien, D. M. Pirone, D. S. Gray, K. Bhadriraju, and C. S. Chen, "Cells lying on a bed of microneedles: an approach to isolate mechanical force," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 4, pp. 1484–1489, 2003.

[30] Tekscan, "FlexiForce Force Sensors," September 2014, http://www.tekscan.com/interface-pressure-measurement-options.

[31] J. I. Bae, T. H. An, Y. S. Kim, and C. K. Ryu, "Analysis of digital load cell using 2.4GHz band's Zig-Bee," in *Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA '08)*, pp. 1358–1361, June 2008.

[32] E. D. Orth, "Semiconductor strain gage pressure transducer," Google Patents, 1972.

[33] P. S. Girão, P. M. P. Ramos, O. Postolache, and J. M. D. Pereira, "Tactile sensors for robotic applications," *Measurement*, vol. 46, no. 3, pp. 1257–1271, 2013.

[34] D. E. Gyi, J. M. Porter, and N. K. B. Robertson, "Seat pressure measurement technologies: considerations for their evaluation," *Applied Ergonomics*, vol. 29, no. 2, pp. 85–91, 1998.

[35] X. Wu, S. Rakheja, and P.-É. Boileau, "Distribution of human-seat interface pressure on a soft automotive seat under vertical vibration," *International Journal of Industrial Ergonomics*, vol. 24, no. 5, pp. 545–557, 1999.

[36] F. Khoshnoud and C. W. de Silva, "Recent advances in MEMS sensor technology-biomedical applications," *IEEE Instrumentation and Measurement Magazine*, vol. 15, no. 1, pp. 8–14, 2012.

[37] H. Zhang and E. So, "Hybrid resistive tactile sensing," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 32, no. 1, pp. 57–65, 2002.

[38] K. Weiß and H. Worn, "The working principle of resistive tactile sensor cells," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 471–476, 2005.

[39] A. M. M. Almassri, W. Z. W. Hasan, S. A. Ahmad, and A. J. Ishak, "A sensitivity study of piezoresistive pressure sensor for robotic hand," in *Proceedings of the IEEE Regional Symposium on Micro and Nano Electronics (RSM '13)*, pp. 394–397, Langkawi, Malaysia, September 2013.

[40] C. M. A. Ashruf, "Thin flexible pressure sensors," *Sensor Review*, vol. 22, no. 4, pp. 322–327, 2002.

[41] G. S. Gupta, S. C. Mukhopadhyay, C. H. Messom, and S. N. Demidenko, "Master-slave control of a teleoperated anthropomorphic robotic arm with gripping force sensing," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 6, pp. 2136–2145, 2006.

[42] M. Ferguson-Pell, S. Hagisawa, and D. Bain, "Evaluation of a sensor for low interface pressure applications," *Medical Engineering and Physics*, vol. 22, no. 9, pp. 657–663, 2000.

[43] C. Steinem, A. Janshoff, and M. A. Cooper, *Piezoelectric Sensors*, Springer, New York, NY, USA, 2007.

[44] P. Puangmali, K. Althoefer, L. D. Seneviratne, D. Murphy, and P. Dasgupta, "State-of-the-art in force and tactile sensing for minimally invasive surgery," *IEEE Sensors Journal*, vol. 8, no. 4, pp. 371–380, 2008.

[45] M. I. Tiwana, S. J. Redmond, and N. H. Lovell, "A review of tactile sensing technologies with applications in biomedical engineering," *Sensors and Actuators A: Physical*, vol. 179, pp. 17–31, 2012.

[46] J. R. Flanagan and A. M. Wing, "Modulation of grip force with load force during point-to-point arm movements," *Experimental Brain Research*, vol. 95, no. 1, pp. 131–143, 1993.

[47] S. B. Lang and S. Muensit, "Review of some lesser-known applications of piezoelectric and pyroelectric polymers," *Applied Physics A*, vol. 85, no. 2, pp. 125–134, 2006.

[48] A. Song, Y. Han, H. Hu, and J. Li, "A novel texture sensor for fabric texture measurement and classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, pp. 1739–1747, 2013.

[49] P. Dario and D. De Rossi, "Tactile sensors and the gripping challenge: increasing the performance of sensors over a wide range of force is a first step toward robotry that can hold and manipulate objects as humans do," *IEEE Spectrum*, vol. 22, no. 8, pp. 46–53, 1985.

[50] A. Song, Y. Han, H. Hu, L. Tian, and J. Wu, "Active perception-based haptic texture sensor," *Sensors and Materials*, vol. 25, no. 1, pp. 1–15, 2013.

[51] V. Maheshwari and R. Saraf, "Tactile devices to sense touch on a par with a human finger," *Angewandte Chemie International Edition*, vol. 47, no. 41, pp. 7808–7826, 2008.

[52] E. Pritchard, M. Mahfouz, B. Evans, S. Eliza, and M. Haider, "Flexible capacitive sensors for high resolution pressure measurement," in *Proceedings of the Sensors Conference*, pp. 1484–1487, Lecce, Italy, 2008.

[53] P. A. Schmidt, E. Maël, and R. P. Würtz, "A sensor for dynamic tactile information with applications in human-robot interaction and object exploration," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 1005–1014, 2006.

[54] B. L. Gray and R. S. Fearing, "A surface micromachined microtactile sensor array," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1–6, Minneapolis, Minn, USA, April 1996.

[55] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers, "Design of the Utah/MIT dextrous hand," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1520–1532, April 1986.

[56] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, and C. Melchiorri, "UBH 3: an anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pp. 4736–4741, May 2004.

[57] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-Hand II: next generation of a dextrous robot hand," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 109–114, 2001.

[58] A. Kargov, T. Asfour, C. Pylatiuk et al., "Development of an anthropomorphic hand for a mobile assistive robot," in *Proceedings of the IEEE 9th International Conference on Rehabilitation Robotics (ICORR '05)*, pp. 182–186, Chicago, Ill, USA, July 2005.

[59] C. Jones, G. Ross, J. R. Hewit, and A. P. Slade, "Robotic sorting of paper items from a random pile," *Mechatronics*, vol. 10, no. 8, pp. 869–880, 2000.

[60] Y. Ma, Z. Cao, X. Dong, C. Zhou, and M. Tan, "A multi-robot coordinated hunting strategy with dynamic alliance," in *Proceedings of the Chinese Control and Decision Conference (CCDC '09)*, pp. 2338–2342, June 2009.

[61] A. Ryberg, M. Ericsson, A.-K. Christiansson, K. Eriksson, J. Nilsson, and M. Larsson, "Stereo vision for path correction in off-line programmed robot welding," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '10)*, pp. 1700–1705, March 2010.

[62] M. Ali, R. Puffer, and H. Roman, "Evaluation of a multifingered robot hand for nuclear power plant operations and maintenance tasks," Tech. Rep., Society of Manufacturing Engineers, 1994.

[63] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.

[64] H. Yussof, J. Wada, and M. Ohka, "Sensorization of robotic hand using optical three-axis tactile sensor: evaluation with grasping and twisting motions," *Journal of Computer Science*, vol. 6, no. 8, pp. 955–962, 2010.

[65] M. T. Mason and J. K. Salisbury Jr., *Robot Hands and the Mechanics of Manipulation*, 1985.

[66] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[67] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.

[68] K. Kim, K. R. Lee, Y. K. Kim et al., "3-Axes flexible tactile sensor fabricated by SI micromachining and packaging technology," in *Proceedings of the 19th IEEE International Conference on Micro Electro Mechanical Systems (MEMS '06)*, pp. 678–681, Istanbul, Turky, January 2006.

[69] T.-R. Hsu, *MEMS & Microsystems: Design, Manufacture, and Nanoscale Engineering*, John Wiley & Sons, New York, NY, USA, 2008.

[70] M. Monta, N. Kondo, and Y. Shibano, "Agricultural robot in grape production system," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2504–2509, May 1995.

[71] H. Kiba, Y. Itoh, and K. Kiryu, "Vehicle body painting robot," Google Patents, 1985.

[72] H.-S. Kang, J. Suh, and S. J. Kwak, "Welding on the fly by using laser scanner and robot," in *Proceedings of the 11th International Conference on Control, Automation and Systems (ICCAS '11)*, pp. 1688–1691, October 2011.

[73] H.-S. Kang, J. Suh, and T.-D. Cho, "Study on robot application technology for laser welding," in *Proceedings of the International Conference on Smart Manufacturing Application (ICSMA '08)*, pp. 354–356, Gyeonggi-do, Republic of Korea, April 2008.

[74] J. G. Webster, *Tactile Sensors for Robotics and Medicine*, John Wiley & Sons, New York, NY, USA, 1988.

[75] R. H. Cannon Jr. and E. Schmitz, "Initial experiments on the end-point control of a flexible one-link robot," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 62–75, 1984.

[76] G. G. Hastings and W. J. Book, "Verification of a linear dynamic model for flexible robotic manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1024–1029, 1986.

[77] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang, "Precision grasp synergies for dexterous robotic hands," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '13)*, pp. 62–67, December 2013.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 471052, 13 pages
http://dx.doi.org/10.1155/2015/471052

Research Article

Hybrid Motion Planning Method for Autonomous Robots Using Kinect Based Sensor Fusion and Virtual Plane Approach in Dynamic Environments

Doopalam Tuvshinjargal, Byambaa Dorj, and Deok Jin Lee

School of Mechanical and Automotive Engineering, Kunsan National University, Gunsan, Jeollabuk 573-701, Republic of Korea

Correspondence should be addressed to Deok Jin Lee; deokjlee@kunsan.ac.kr

Received 5 February 2015; Accepted 27 March 2015

Academic Editor: Guangming Song

A new reactive motion planning method for an autonomous vehicle in dynamic environments is proposed. The new dynamic motion planning method combines a virtual plane based reactive motion planning technique with a sensor fusion based obstacle detection approach, which results in improving robustness and autonomy of vehicle navigation within unpredictable dynamic environments. The key feature of the new reactive motion planning method is based on a local observer in the virtual plane which allows the effective transformation of complex dynamic planning problems into simple stationary in the virtual plane. In addition, a sensor fusion based obstacle detection technique provides the pose estimation of moving obstacles by using a Kinect sensor and a sonar sensor, which helps to improve the accuracy and robustness of the reactive motion planning approach in uncertain dynamic environments. The performance of the proposed method was demonstrated through not only simulation studies but also field experiments using multiple moving obstacles even in hostile environments where conventional method failed.

1. Introduction

The capability of mobile robots to autonomously navigate and safely avoid obstacles plays a key role for many successful real-world applications [1]. To date, a major research work has been applied to analyze and solve the motion planning in a completely known environment with largely static or, to some extent, moving obstacles. Motion planning in dynamic environments is still among the most difficult and important problems in mobile robotics. The autonomous motion planning approaches for the robots can be classified into three different paradigms such as the hierarchical, reactive, and hybrid approach [2]. These paradigms in robot navigation community point out to a major dichotomy classified into two categories: planned based approach and behavior based technique. The hierarchical (or planned based) navigation approaches have a serial control architecture with which robots sense the known world, plan their operations, and act to follow a path expressed in global coordinates based on this sensed model. For instance, deterministic and probabilistic roadmap methods are widely used in [2–4]; potential field based methods are suggested in [5, 6]. In [7], a collision-free path planning approach was suggested based on Bezier curves. A novel optimization method considering robot posture and path smoothness is presented in [8]. Since there is no direct connection between the sensing and acting, the robot is limited to operate only in static environment. In [9], a path planning based robot navigation approach was proposed to cope with unexpected changing environment using D^* approach and automatic docking system for recharging home surveillance robot system is proposed in [10], but the performance is limited when obstacles are allowed to move in the workspace. The feature of the planned based approaches makes the robot difficult to manage to interact with a constantly changing dynamic environment while performing complex tasks at slow speed.

On the other hand, unlike the preceding methods, the behavior based approaches [11–18] or called reactive based methods utilize local control laws relative to local features and rely on accurate local feature detection to cope with these unexpected chances in a reactive way. Reactive navigation differs from the planned navigation approach in the sense

that when a mission is assigned or a goal location is given, the robot does not plan its path but rather navigate itself by reacting to its immediate environment in real time. The main idea of the reactive paradigm is to separate the control system into small units of sensor-action pairs with a layered modular architecture resulting in fast execution of the control algorithm [12]. There are other types of developments in local reactive path planning approaches, such as Vector Field Force (VFF) [13] and Vector Field Histogram (VFH) [14]. The VFF and VFH methods generate histograms from senor data in order to generate control commands for the vehicle but they do not take into account the dynamic and kinematic constraints of the vehicle.

However, there have been a few reactive works that utilize the kinematic or dynamic information of the environment to compute the motion commands for avoiding unexpected changes in the environment. When the velocity information of the objects obtained from available sensors is utilized, the robot navigation system can compute trajectories resulting in improving the motion performance regarding other obstacle avoidance methods [15–19]. The curvature velocity method (CVM) [15] and the dynamic windows approach (DWA) [16] search an appropriate control command in velocity space by maximizing an objective function which has criteria such as speed, distance between obstacles, and remaining distance towards the final destination. The CVM and DWA method, however, could increase the order of complexity resulting from the optimization of the cost function. In [17–19], a velocity information based approach for navigation and collision detection based on the kinematic equation is introduced by using the notion of collision cones in the velocity space. In a similar way, the concept of velocity obstacles [20, 21] takes the velocity of the moving obstacles into account, which results in a shift of the collision cones. This method is restricted to obstacles with linear motion, and thus the nonlinear velocity obstacle approach is introduced to extend to cope with obstacles moving along arbitrary trajectories [22]. The key concept of velocity obstacles is to transform the dynamic problem into several static problems in order to increase the capability of avoiding dynamic obstacle within unexpected environment changes [23]. Meanwhile, sensor based motion planning techniques are also widely used for robot navigation applications in dynamic environments, where the pose estimates of the moving obstacles are obtained by using sensory systems [24–26]. These sensor based navigation approaches also require the knowledge of the obstacle's velocities for an accurate navigation solution.

In this work, a new sensor fusion based hybrid reactive navigation approach for autonomous robots is proposed in dynamic environments. The contribution of the new motion planning method lies on the fact that it integrates a local observer in a virtual plane as a kinematic reactive path planner [23] with a sensor fusion based obstacle detection approach which can provide a relative information of moving obstacles and environments, resulting in an improved robustness and accuracy of the dynamic navigation capability. The key feature of the reactive motion planning method is based on a local observer in the virtual plane approach which makes the effective transformation of complex dynamic planning problems into simple stationary ones along with a collision cone in the virtual plane approach [23]. On the other hand, a sensor fusion based planning technique provides the pose estimation of moving obstacles by using sensory systems and thus it could improve the accuracy, reliability, and robustness of the reactive motion planning approach in uncertain dynamic environments. The hybrid reactive planning method allows an autonomous vehicle to reactively change heading and velocity to cope with an obstacle around in each planning time. As a sensory system, Microsoft Kinect device [27] which could obtain distance between the camera and target objects is utilized. The advantage of using Kinect is on its capability of calculating the distance between two objects on the world coordinate frame. In case that the two objects are placed closer, a sonar sensor mounted on the robot can detect and make a precise distance calculation in combination with the Kinect sensor data. The integrated hybrid motion planning with the integration of the virtual plane approach and sensor based estimation method allows the robot to find the appropriate windows for the speed and orientation to move with a collision-free path in dynamic environments, making its usage very attractive and suitable for real-time embedded applications. In order to verify the performance of the suggested method, real experiments are carried out for the autonomous navigation of a mobile robot in the dynamic environments using multiple moving obstacles. Here two mobile robots act on the moving obstacles and one has to avoid collision with the other robot.

The rest of the work is organized as follows. In Section 2 we introduce the kinematic equations and the geometry of the dynamic motion planning problem. In Section 3, the concept of the hybrid reactive navigation using virtual plane approach is given. The configuration and system architecture of the Kinect device is discussed in Section 4. Simulation and experimental tests are shown and discussed in Section 5.

2. Definition of Dynamic Motion Planning

In this section, the relative velocity obstacle based motion planning algorithms for collision detection and control laws are defined [23]. Figure 1 shows some geometry parameters for the navigation in dynamic environment for the mobile robot. The world is attached to a global fixed reference frame of coordinates $\{W\}$, and its origin point is the origin O. It is possible to attach local reference frames to every moving object in the working space. The suggested method is a reactive navigation method with which the robot needs to change the path to avoid either moving or static obstacles within a given radius, that is, the coverage area (CA).

The line of sight of the robot l_r is the imaginary straight line that starts from the origin and is directed toward the reference center point of the robot R. The line-of-sight angle θ_r is the angle made by the sight l_r. The distance l_{gr} between robot R and the goal G is calculated by

$$l_{gr} = \sqrt{\left(x_g - x_r\right)^2 + \left(y_g - y_r\right)^2}, \tag{1}$$

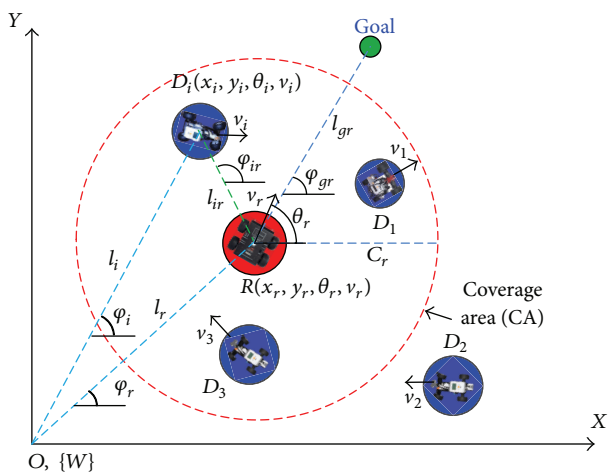


Figure 1: Geometry of the navigation problem. Illustration of the kinematic and geometric variables.

where (x_g, y_g) is the coordinates of the final goal point and (y_r, x_r) is the state of the robot in $\{W\}$. The mobile robot has a differential driving mechanism using two wheels and the kinematic equation of the wheeled mobile robot can be given by

$$\begin{aligned} \dot{x}_r &= v_r \cos\theta_r, \\ \dot{y}_r &= v_r \sin\theta_r, \\ \dot{v}_r &= a_r, \\ \dot{\theta}_r &= w_r, \end{aligned} \tag{2}$$

where a_r is the robot's linear acceleration and v_r and w_r are the linear and angular velocities. (θ_r, v_r) are the control inputs of the mobile robot. The line-of-sight angle φ_{gr} which is obtained from the angle made by the line of sight l_{gr} is given by the following equations:

$$\begin{aligned} \cos\varphi_{gr} &= \frac{x_g - x_r}{\sqrt{(x_g - x_r)^2 + (y_g - y_r)^2}} \\ \tan\varphi_{gr} &= \frac{y_g - y_r}{x_g - x_r}. \end{aligned} \tag{3}$$

Now, the kinematic equation of the ith obstacle D_i is expressed by

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i, \\ \dot{y}_i &= v_i \sin\theta_i, \\ \dot{\theta}_i &= w_i, \end{aligned} \tag{4}$$

where the obstacle has the linear velocity v_i and the angular velocities w_i, and θ_i is the orientation angle. The Euclidian distance of the line of sight l_{ir} between the robot and the ith obstacle is calculated by

$$l_{ir} = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \tag{5}$$

and the line-of-sight angle φ_{ir} is expressed by

$$\tan\varphi_{ir} = \frac{y_i - y_r}{x_i - x_r}. \tag{6}$$

The evolution of the range and turning angle between the robot and an obstacle for dynamic collision avoidance is computed by using the tangential and normal component of the relative velocity in the polar coordinates as follows:

$$\begin{aligned} \dot{l}_{ir} &= v_i \cos(\theta_i - \varphi_{ir}) - v_r \cos(\theta_r - \varphi_{ir}) \\ l_{ir}\dot{\varphi}_{ir} &= v_i \sin(\theta_i - \varphi_{ir}) - v_r \sin(\theta_r - \varphi_{ir}). \end{aligned} \tag{7}$$

From these equations it is shown that a negative sign of $\dot{l}_{ir}$ indicates that the robot is approaching obstacle D_i, and if the rate is zero, the range implies constant distance between the robot and the obstacle. Meanwhile, a zero rate for the line-of-sight angle indicates the motion of D_i is a straight line. The relative polar system presents a simple but very effective model that allows real-time representation of the relative motion between the robot and moving obstacle [23].

3. Hybrid Reactive Motion Planning Approach

3.1. Virtual Plane Based Reactive Motion Planning. In this section, the virtual plane method which allows transforming a moving object of interest into a stationary object is briefly reviewed [23]. The transformation used in the virtual plane is achieved by introducing a local observer that allows the robot to find the appropriate windows for the speed and orientation to move in a collision-free path. Through this transformation, the collision course between the robot R and the ith obstacle D_i is reduced to a collision course between the virtual robot R^v and the initial position $D_i(t_0)$ of a real obstacle. The components of the relative velocity between R^v and $D_i(t_0)$ along and across $\dot{l}_{ir}$ are given by

$$\begin{aligned} \dot{l}_{ir} &= -v_r^v \cos(\theta_r^v - \varphi_{ir}) \\ l_{ir}\dot{\varphi}_{ir} &= -v_r^v \sin(\theta_r^v - \varphi_{ir}), \end{aligned} \tag{8}$$

where v_r^v and θ_r^v are the linear velocity and orientation of the virtual robot. The linear velocity and orientation angle of R^v can be written as follows:

$$v_r^v = \sqrt{(\dot{x}_i - \dot{x}_r)^2 + (\dot{y}_i - \dot{y}_r)^2} \tag{9}$$

$$\tan\theta_r^v = \frac{\dot{y}_i - \dot{y}_r}{\dot{x}_i - \dot{x}_r}. \tag{10}$$

Note that the tangential and normal equations given in (7) for the dynamic motion planning are rewritten in terms of the virtual robot as an observer, leading to a stationary motion planning problem. More details concerning the virtual planning method can be referred to in [23].

Collision detection is expressed in the virtual plane, but the final objective is to make the robot navigate toward the

goal with collision-free path in the real plane. The orientation angle of the robot in the real plane is calculated by

$$\dot{x}_r = \dot{x}_r^v + \dot{x}_i$$
$$\dot{y}_r = \dot{y}_r^v + \dot{y}_i. \tag{11}$$

This is the inverse transformation mapping the virtual plane into the real plane and gives the velocity of the robot as a function of the velocities of the virtual robot and the moving object. The speed and orientation of the real robot can be computed from the virtual robot and the moving object velocities as follows:

$$v_r = \sqrt{\left(\dot{x}_r^v + \dot{x}_i\right)^2 + \left(\dot{y}_r^v + \dot{y}_i\right)^2}$$
$$\tan\theta_r = \frac{\dot{y}_r^v + \dot{y}_i}{\dot{x}_r^v + \dot{x}_i}. \tag{12}$$

3.2. Navigation Laws. In order to make the robot navigate toward the final goal, a kinematic based linear navigation law is used as [23]

$$\theta_r = M\varphi_{gr} + c_1 + c_0 e^{-at}, \tag{13}$$

where φ_{gr} is the line-of-sight angle of the robot final goal, and the variables are deviation terms characterizing the final desired orientation angle of the robot and indicating the initial orientation of the robot. The term M is a navigation parameter with $M > 1$, and a is a given positive gain. On the other hand, the collision course in the virtual plane with $D_i(t_0)$ is characterized by

$$\theta_r^v \in \mathrm{CCVP}_i. \tag{14}$$

The collision cone in the virtual plane (CCVP) is given by

$$\mathrm{CCVP}_i = \left[\varphi_{ir} - \beta_i, \varphi_{ir} + \beta_i\right], \tag{15}$$

where β_i is the angle between the lines of the upper and lower tangent limit points in D_i. The direct collision course between R and D_i is characterized by

$$\tan\theta_r^v = \frac{\dot{y}_i - \dot{y}_r}{\dot{x}_i - \dot{x}_r} = \left[\tan\left(\varphi_{ir} - \beta_i\right), \tan\left(\varphi_{ir} + \beta_i\right)\right]. \tag{16}$$

After the orientation angle θ_r^v of the virtual robot is computed in terms of the linear velocity of the robot and the moving obstacles as given in (10), it is possible to write the expressions of the orientation angle θ_r and the speed v_r for the real robot controls v_r or in terms of the linear velocity and orientation angle of the moving obstacle and the virtual robot as follows:

$$v_r = \frac{v_i\left(\tan\theta_r^v \cos\theta_i - \sin\theta_i\right)}{\tan\theta_r^v \cos\theta_r - \sin\theta_r}$$
$$\theta_r = \theta_r^v - \arcsin\left[\frac{v_i \sin\left(\theta_r^v - \theta_i\right)}{v_r}\right]. \tag{17}$$

For the robot control, the desired value of the orientation angle in the virtual plane can be expressed based on using the linear navigation law as

$$\theta_r^{v^*}(t) = \alpha_{i,k} + c_1 + c_0 \exp\left\{-a\left(t - t_d\right)\right\}, \quad k = 1, 2, \tag{18}$$

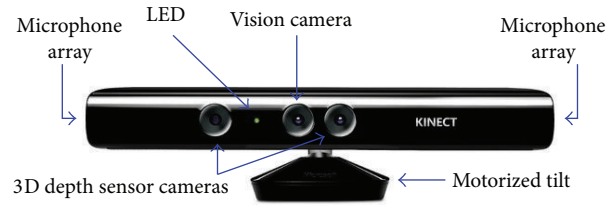


Figure 2: Architecture of Microsoft Kinect sensor.

where t_d denotes the time when the robot starts deviation for collision avoidance, and $\alpha_{i,1}$ and $\alpha_{i,2}$ are the left and right line-of-sight angles between the reference deviation points and the points on the collision cone in the virtual plane. Finally, based on the desired orientation in the virtual plane, the corresponding desired speed value v_r^* for the robot is calculated by

$$v_r^* = \frac{v_i\left(\tan\theta_r^{v^*} \cos\theta_i - \sin\theta_i\right)}{\tan\theta_r^{v^*} \cos\theta_r - \sin\theta_r}. \tag{19}$$

In a similar way, the corresponding desired orientation value can be expressed by

$$\tan\theta_r^* = \frac{v_r^v \sin\left(\theta_r^{v^*}\right) + v_i \sin\left(\theta_i\right)}{v_r^v \cos\left(\theta_r^{v^*}\right) + v_i \cos\left(\theta_i\right)}. \tag{20}$$

Note that, for the robot navigation including a collision avoidance technique within dynamic environments, either the linear velocity control expressed in (19) or the orientation angle control in (20) can be utilized.

3.3. Sensor Fusion Based Range and Pose Estimation. Low-cost range sensors are an attractive alternative to expensive laser scanners in application areas such as motion planning and mapping. The Microsoft Kinect [26] is a sensor which consists of an IR sensor, an IR camera, an RGB camera, a multiarray microphone, and an electrical motor, providing the tilt function to the sensor (shown in Figure 2). The Kinect sensor captures not only depth but also color images simultaneously at a frame rate of up to 30 fps. Some key features are illustrated in [26–29]. The RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter at a frame rate 30 Hz. The monochrome depth sensing video stream has a VGA resolution (640 × 480 pixels) with 11-bit depth, which provides 2048 levels of sensitivity. Depth data is acquired by the combination of IR projector and IR camera. The microphone array features four microphone capsules and operates with each channel processing 16-bit audio at a sampling rate of 16 kHz. The motorized pivot is capable of tilting the sensor up to 27° either up or down.

The features of Kinect device make its application very attractive to autonomous robot navigation. In this work, the Kinect sensor is utilized for measuring range to moving

Table 1: Kinect's focal length and field of view.

Camera	Focal length (pixel)	Field of view (degrees)	
		Horizontally	Vertically
RGB	525	63	50
IR	580	57	43

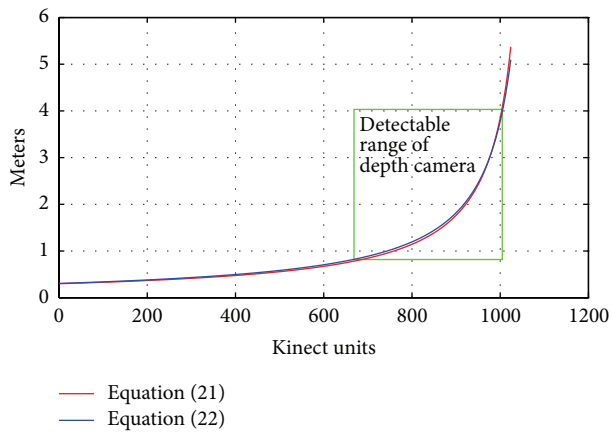


Figure 3: Kinect depth measurement and actual distance.

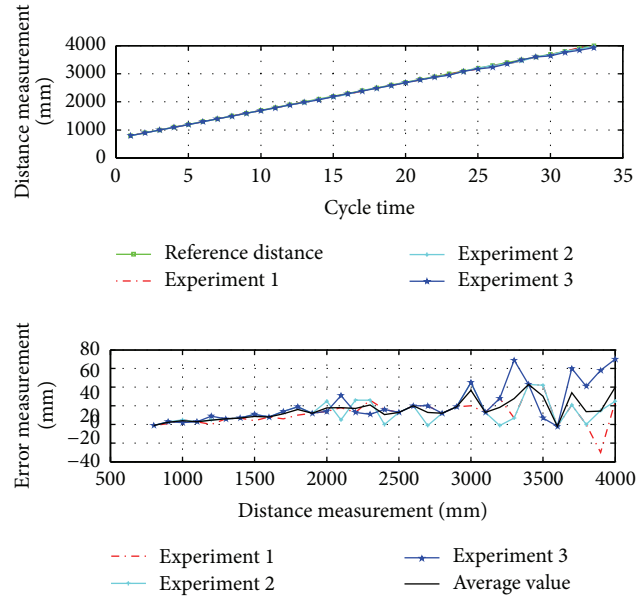


Figure 4: Kinect depth camera measurement experiment and error.

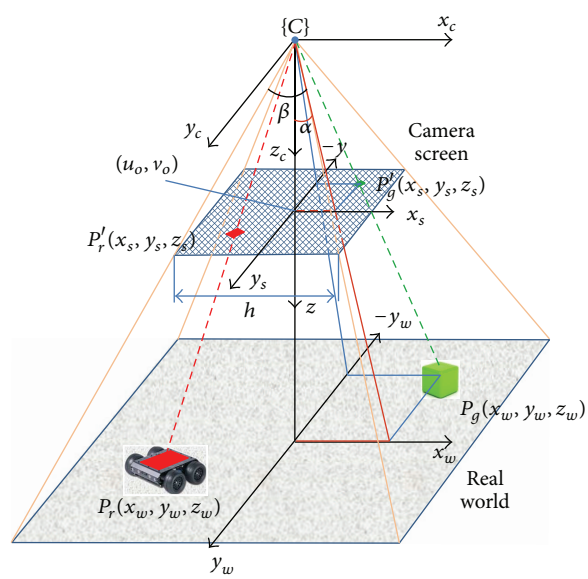


Figure 5: Robot and obstacle localization using the Kinect sensor for ranging and positioning computation.

obstacles and estimating color-based locations of objects for dynamic motion planning.

Before going into detail, the concept of the calculation of the real coordinates is discussed. Kinect camera has some good advantages such as depth sensor with minimum 800 mm and maximum range 4000 mm. Camera focus is constant and given and thus real distance between camera and chosen target is easily calculated. The parameters used in Kinect sensor are summarized in Table 1.

Two similar equations have been proposed by researcher, where one is based on the function $1/D_{\text{value}}$ and the other is using $\tan(D_{\text{value}})$. The distance between a camera and a target object z_w is expressed by

$$z_w = \frac{1}{(D_{\text{value}} \times (-0.0030711016) + 3.3309495161)} \quad (21)$$

$$z_w = 0.1236 \times \tan\left(\frac{D_{\text{value}}}{2842.5 + 1.1863}\right). \quad (22)$$

Figure 3 shows the detectable ranges of a depth camera where the distances in world coordinate based on the above two equations are computed by limiting the raw depth to 1024 that corresponds to about 5 meters.

Figure 4 shows the error results of distance measurement experiments using a Kinect's depth camera. In this experiment, the measured distance using a ruler is noted by green which gives a reference distance, and three repetitive experiments are carried out and they are drawn in red, light blue, and blue colors. From the experiment, it is shown that the errors of the depth measurements from the Kinect sensor are proportional to the distance.

Figure 5 shows a general schematics of geometric approach to find the x and y coordinates using the Kinect sensor system, where h is the screen height in pixels and β is the field of view of the camera. The coordinates of a point on the image plane of the robot P_r' and goal P_g' are transformed into the world coordinates P_r and P_g, and it is calculated by

$$P = (x_w, y_w) \longrightarrow P' = (x_s, y_s, z_s)$$
$$x_w = \frac{x_s}{z_w}, \qquad y_w = \frac{y_s}{z_w}. \quad (23)$$

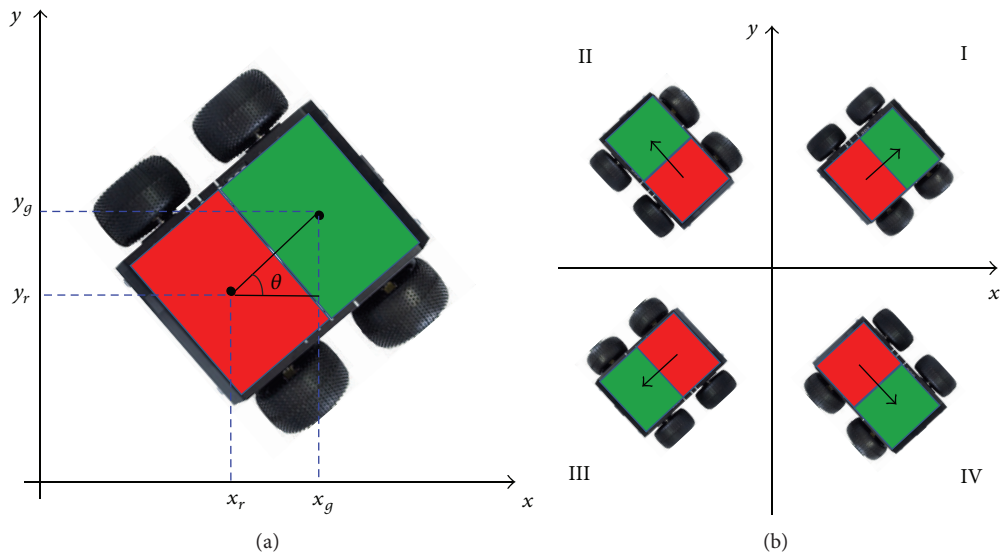


FIGURE 6: Robot heading angle computation approach.

Each of coordinates x_w, y_w, z_w of two red (robot) and green (goal) points is used as the input into the vision system, and P' is computed by

$$\begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}, \tag{24}$$

where f is the focal length of the camera. P' is calculated at the pixel coordinates divided by ρ_u and ρ_v, and the values of the pixel of the image u and v are the pixel coordinates and they are obtained from the following equations:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix}$$

$$P' = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{u'}{w'} \\ \frac{v'}{w'} \end{pmatrix}, \qquad \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = C \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}. \tag{25}$$

In the experiment, the final goal and robots are recognized by a built-in RGB camera. In addition, the distance of an object is measured within mm accuracy using IR camera, and the target object's pixel coordinates (x_s, y_s) are estimated by using a color-based detection approach. In this work, the distance measured between the planning field and the Kinect sensor is 2700 mm which becomes the depth camera's detectable range. The horizontal and vertical coordinates of an object are calculated as follows:

(1) horizontal coordinate:

$$\alpha = \arctan\left(\frac{x_s}{f}\right)$$
$$x_w = z_w \times \sin\alpha; \tag{26}$$

(2) vertical coordinate:

$$\alpha = \arctan\left(\frac{y_s}{f}\right)$$
$$y_w = z_w \times \sin\alpha, \tag{27}$$

where z_w is the distance to the object obtained from Kinect sensor. Now, those real-world coordinates obtained in the above are used in dynamic path planning procedures.

In general, the camera intrinsic and extrinsic parameters of the color and depth cameras are set as default values, and thus it is necessary to calibrate them for accurate tests. The calibration of the depth and RGB color cameras in the Kinect sensor can be applied by using a mathematical model of depth measurement and creating a depth annotation of a chessboard by physically offsetting the chessboard from its background, and details of the calibration procedures can be referred to in [30, 31].

As indicated in the previous section, for the proposed relative velocity obstacle based dynamic motion planning, the accurate estimates of the range and orientation of an object play an important role. In this section, an efficient new approach is proposed to estimate the heading information of the robot using a color detection approach. First, the robot is covered by green and red sections shown in Figure 6.

Then, using a color detection method [32] the center location of the robot is calculated, and after finding denominate

```
Input: Coordinate of Robot, obstacle 1, obstacle 2 and goal
Output: Speed of robot's right and left wheels
Calculate l_gr using Kinect
While l_gr > 0 do
  Calculate l_gr and φ_gr
  Send robot speed
  if All D_i in CA
    Calculate l̇_ir using Kinect
    if All l̇_ir > 0 then
      There is no collision risk, keep send robot speed
    else
      Construct the virtual plane
      Test the collision in the virtual plane
      if there is a collision risk then
        Check sonar sensor value
        if sonar value is too small then
          Chose θ_r of quick motion control
          Send robot speed
        else
          Construct the θ-window
          Chose the appropriate values for θ_r
          Send robot speed
      end if
    end if
  end if
end while
```

Algorithm 1: Hybrid reactive dynamic navigation algorithm.

heading angle θ as shown in (28), new heading information in each four different phase sections is computed by using the following equations:

$$\begin{aligned} \Delta x &= x_g - x_r \\ \Delta y &= y_g - y_r \\ \theta &= \tan^{-1}\left(\frac{y_g - y_r}{x_g - x_r}\right) \end{aligned} \tag{28}$$

$$\begin{aligned} &(1)\ x_g > x_r, \quad y_g > y_r \\ &\qquad \theta = \theta + 0 \\ &(2)\ x_g < x_r, \quad y_g > y_r \\ &\qquad \theta = 3.14 - \theta \\ &(3)\ x_g < x_r, \quad y_g < y_r \\ &\qquad \theta = 3.14 + \theta \\ &(4)\ x_g > x_r, \quad y_g < y_r \\ &\qquad \theta = 6.28 - \theta. \end{aligned} \tag{29}$$

Finally, the relative velocity obstacle based reactive dynamic navigation algorithm with the capability of collision avoidance is summarized in Algorithm 1.

4. Experimental Results

For the evaluation and verification of the proposed sensor based reactive dynamic navigation algorithms, both simulation study and experimental tests are carried out with a realistic experimental setup.

4.1. Experimental Scenario and Setup. For experimental tests, two robots are assigned as moving obstacles and the third one is used as a master robot that generates control commands to avoid the dynamic obstacles based on the suggested reactive motion planning algorithms. For the moving obstacles, two NXT Mindstorm based vehicles that can either move in a random direction or follow a designated path are developed. The HBE-RoboCAR equipped with ultrasonic and encoder sensors is used as a master robot as shown in Figure 7.

The HBE-RoboCAR [33] has 8-bit AVR ATmega128L processor. The robot is equipped with multiembedded processor modules (embedded processor, FPGA, MCU). It provides detection of obstacles with ultrasonic and infrared sensor, motion control with acceleration sensor, and motor encoder. HBE-RoboCAR has the ability to communicate with other device either wireless or wired technology such as Bluetooth module and ISP, UART interfaces, respectively. In this work, HBE-RoboCAR is connected to a computer on the ground control station using Bluetooth wireless communication. Figure 7 shows the hardware specification and sensor systems for the robot platform, and Figure 8 shows the interface and control architecture for the embedded components of HBE-RoboCAR [33].

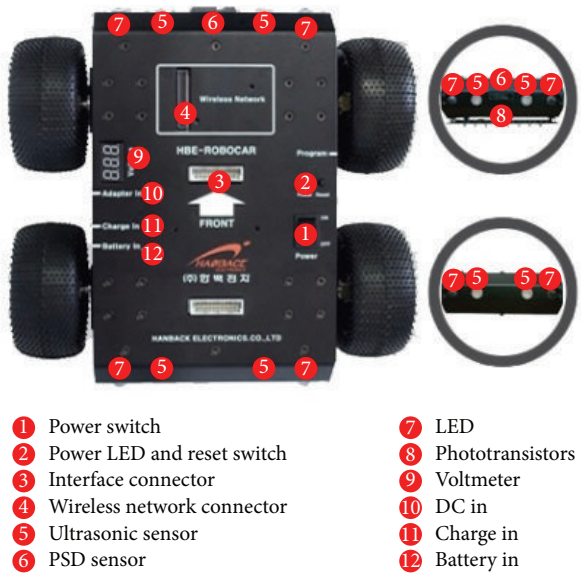


Figure 7: Mobile robot hardware and sensor systems (HBE-RoboCAR [33]).

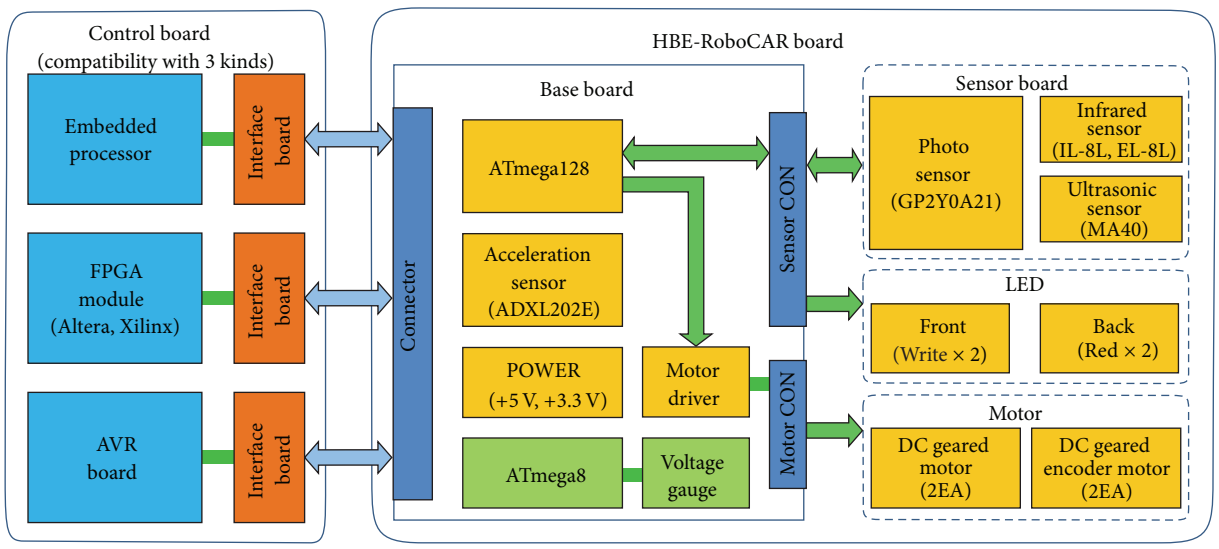


Figure 8: Block diagram of the HBE-RoboCAR embedded system.

For the dynamic obstacle avoidance, the relative velocity obstacle based navigation laws require the range and heading information from sensors. For the range estimation, Kinect sensor is utilized. If the Kinect sensor detects the final goal using a color-based detection algorithm [32, 34, 35], it sends the information to the master robot. After receiving the target point, the master robot starts the onboard navigation algorithm to reach the goal while avoiding dynamic obstacles. When the robot navigates in the experimental field, the distance to each moving obstacle is measured by the Kinect sensor and the range information is fed back to the master robot via Bluetooth communication as inputs to the reactive motion planning algorithms. The detailed scenario for the experimental setup is illustrated in Figure 9.

4.2. Simulation Results. Figure 10 shows the simulation results of the reactive motion planning on both the virtual plane and the real plane. In this simulation, the trajectories of two obstacles were described by the blue and black color lines, the trajectory of the master robot was depicted in the red line, and the goal was indicated by green dot. As can be clearly seen in the real plane and the virtual plane in Figures 10(b) and 10(a), the master robot avoided the first obstacle which was moving into the master robot and successfully reached the target goal after avoiding the collision with the second moving obstacle just before reaching the target. While the master robot avoids the obstacles, it generates a collision cone by choosing a deviation point on the virtual plane. On the virtual plane, the radius of the collision cone

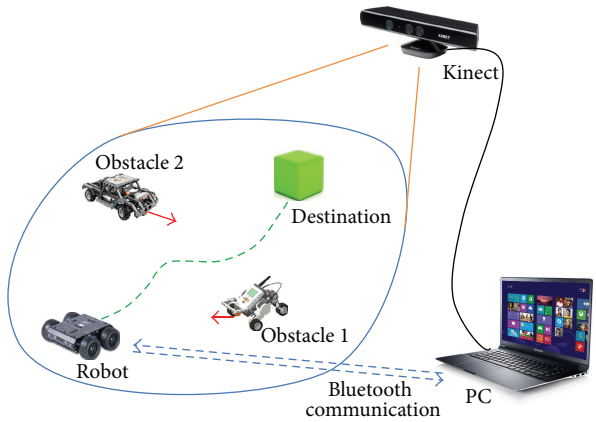
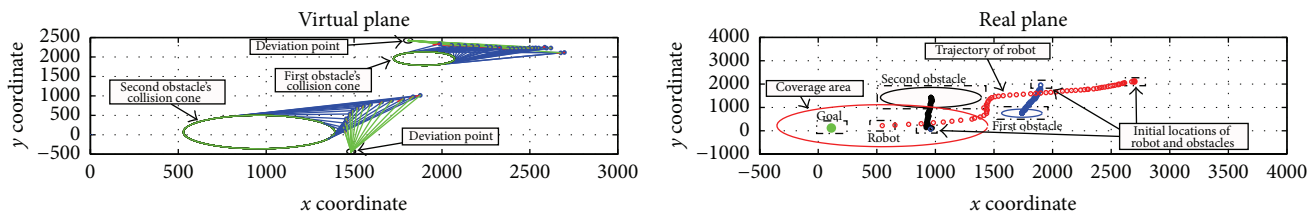


Figure 9: Test scenario and architecture for experimental setup.



(a) (b)

Figure 10: Simulation results on virtual plane and real plane.

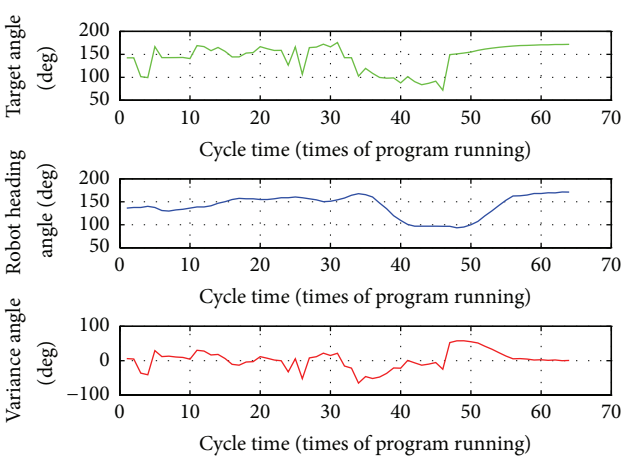


Figure 11: Orientation angle information: target angle, robot heading angle, and variance angle.

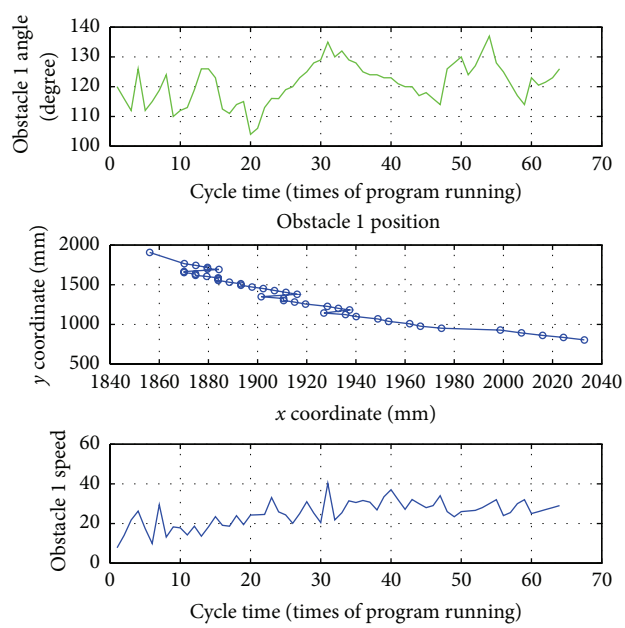


Figure 12: Results of first moving obstacle's orientation angle, speed, and position in x-y coordinates.

is the same as the obstacle's one, and the distance between the deviation point and the collision cone is dependent on the radius of the master robot. The ellipses indicate the initial locations of the robot and the obstacles.

In Figure 11, the orientation angle information used for the robot control was illustrated. The upper top plot showed the angle of the moving robot to the target from the virtual plane, the second plot showed the robot heading angle commanded for the navigation control, and the third plot showed the angle difference between the target and robot heading angle. At the final stage of the path planning, the commanded orientation angle and the target angle to the goal point become the same. Instead of controlling the robot with the orientation angle, the speed of the master robot can be used to avoid the moving obstacles.

Figures 12 and 13 show each moving obstacle's heading angle, linear velocity, and trajectory. As can be seen, in order

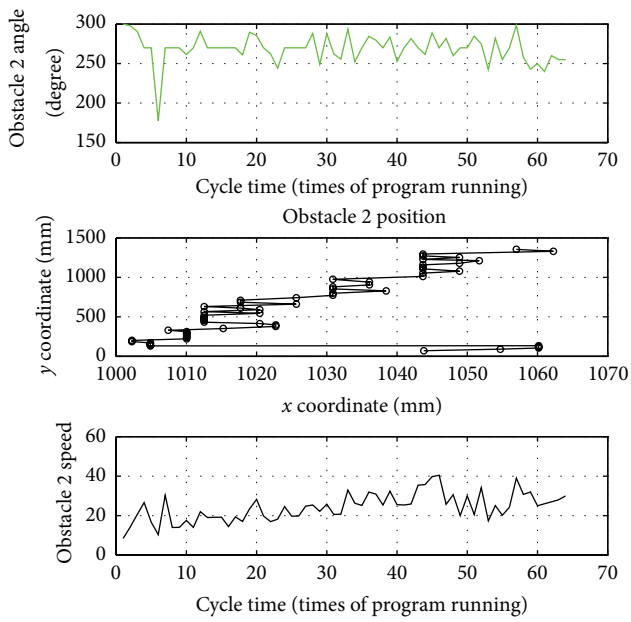


Figure 13: Results of second moving obstacle's orientation angle, speed, and position in x-y coordinates.

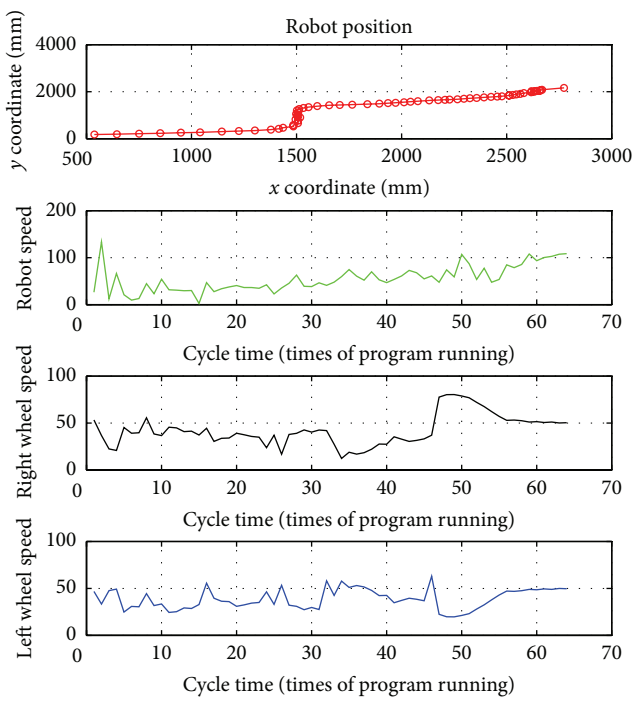


Figure 14: Results of robot's trajectory, speed, and right and left wheels speed.

to carry out a dynamic path planning experiment, the speed and the heading angle were changed during the simulation, resulting in uncertain cluttered environments.

Figure 14 shows the mobile robot's trajectory from the start point to the goal point, and also the forward velocity and each wheel speed from encoder. As can be seen, the trajectory of the robot has a sharp turn around the location (1500 mm, 1000 mm) in order to avoid the second moving obstacle. It is seen that the right and left wheel speeds are mirrored along a time axis. Also, we can see relationship of the variance angle and robot's right and left wheels speed.

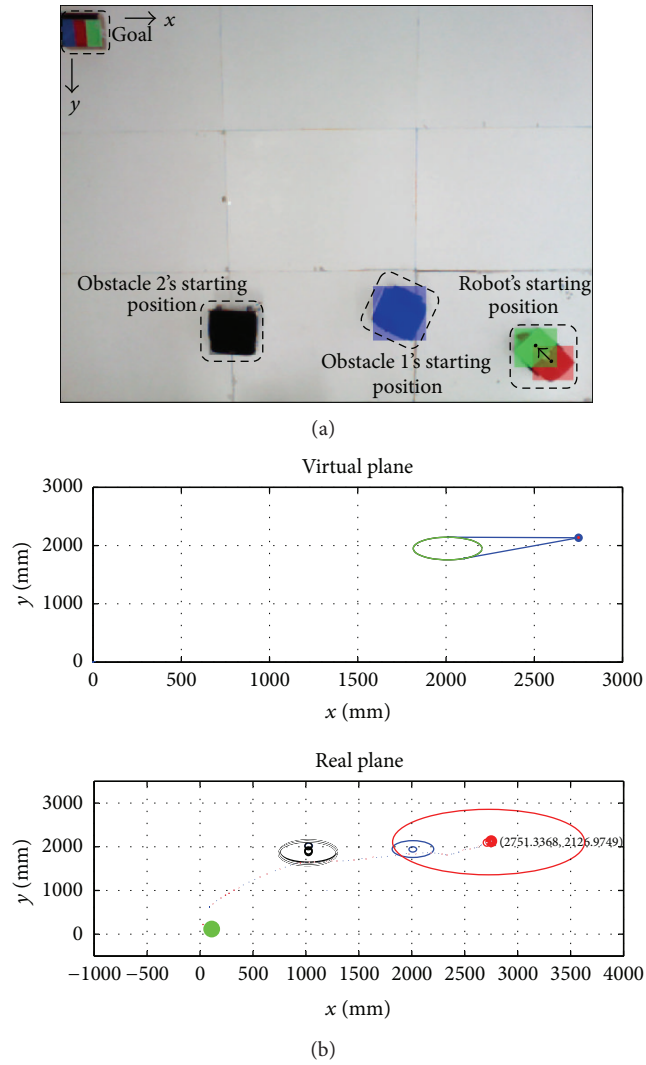


Figure 15: (a) Experiment environment (initial stage). (b) Plots of the experimented data at initial stage.

4.3. Field Test and Results. Further verification of the performance of the proposed hybrid dynamic path planning approach for real experiments was carried out with the same scenario used in the previous simulation part. In the experiment, two moving obstacles are used and a master robot moves without any collision with the obstacles to the target point as shown in Figure 15(a), and the initial locations of the obstacles and the robot are shown in Figure 15(b). The red dot is the initial starting position of the master robot at (2750 mm, 2126 mm), the black dot is the initial location of the second obstacle at (2050 mm, 1900 mm), and the blue dot is the initial location of the first moving obstacle at (1050 mm, 2000 mm). In the virtual plane, the collision cone of the first obstacle is depicted as shown in the top plot of Figure 15(b),

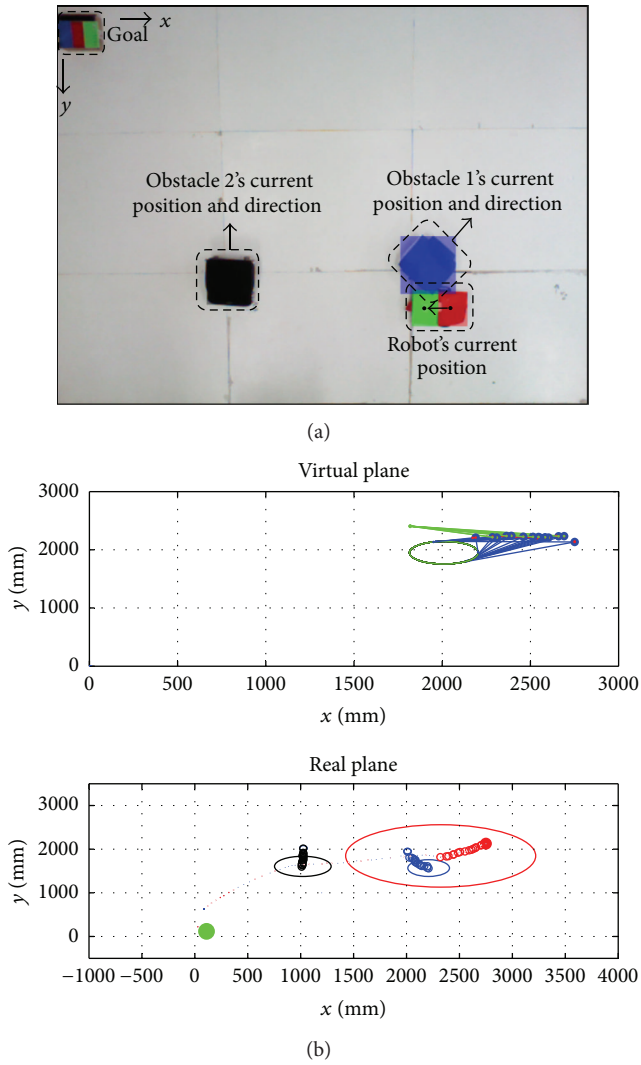


Figure 16: (a) Experiment result of collision avoidance with the first obstacle. (b) Plot of the trajectories of the robot and the obstacles in the virtual and real planes during the collision avoidance with the first moving obstacle.

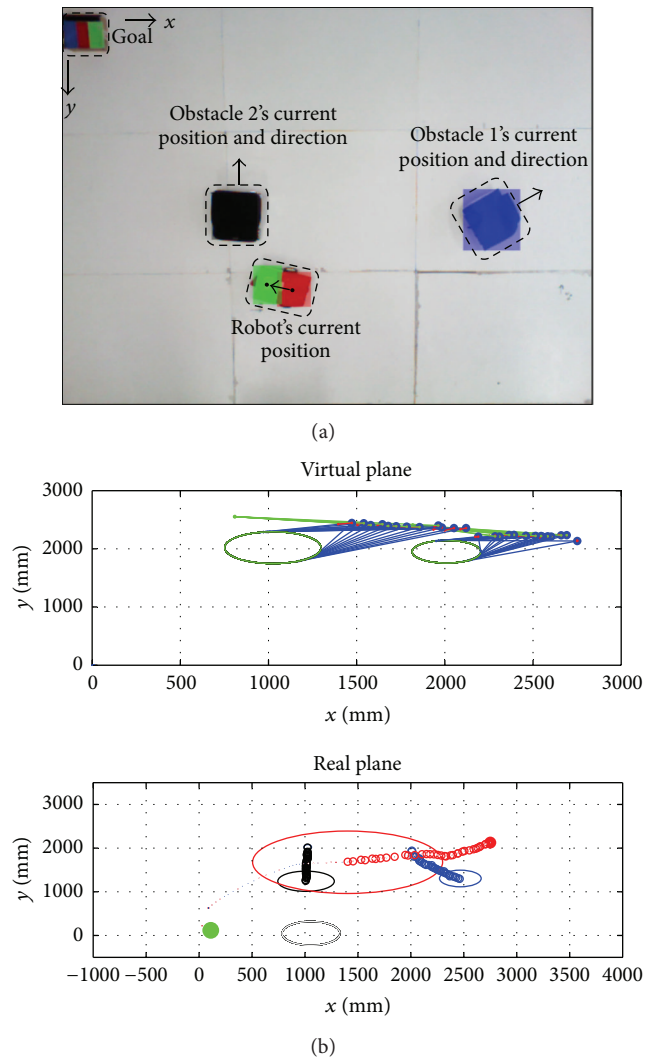


Figure 17: (a) Experiment result of collision avoidance with the second obstacle. (b) Plot of the trajectories of the robot and the obstacles in the virtual and real planes during the collision avoidance with the second moving obstacle.

and the robot carries out its motion control based on the collision cone in the virtual plane until it avoids the first obstacle.

Figure 16 showed the collision avoidance performance with the fist moving obstacle, and as can be seen, the master robot avoided the commanded orientation control into the left direction without any collisions, which is described in detail in the virtual plane in the top plot of Figure 16(b). The trajectory and movement of the robot and the obstacles were depicted in the real plane in Figure 16(b) for the detailed analysis.

In a similar way, Figure 17(a) illustrated the collision avoidance with the second moving obstacle, and the detailed path and trajectory are described in Figure 17(b). The top plot of Figure 17(b) shows the motion planning in the virtual plane, where the initial location of the second moving obstacle is recognized at the center of (2050 mm, 1900 mm). Based on this initial location, the second collision cone is constructed with a big green ellipse that allows the virtual robot to navigate without any collision with the second obstacle. The trajectory of the robot motion planning in the real plane is depicted in the bottom plot of Figure 17(b).

Now, at the final phase after avoiding all the obstacles, the master robot reached the target goal with a motion control as shown in Figure 18(a). The overall trajectories of the robot from the starting point to the final goal target in the virtual plane were depicted in the top plot of Figure 18(a), and the trajectories of the robot in the real plane were depicted in the bottom plot of Figure 18(b). Note that the trajectories of the robot location differ from each other in the virtual plane and the real plane. However, the orientation change gives the same direction change of the robot in both the virtual and the real plane. In this plot, the green dot is the final goal point and the robot trajectory is depicted with the red dotted

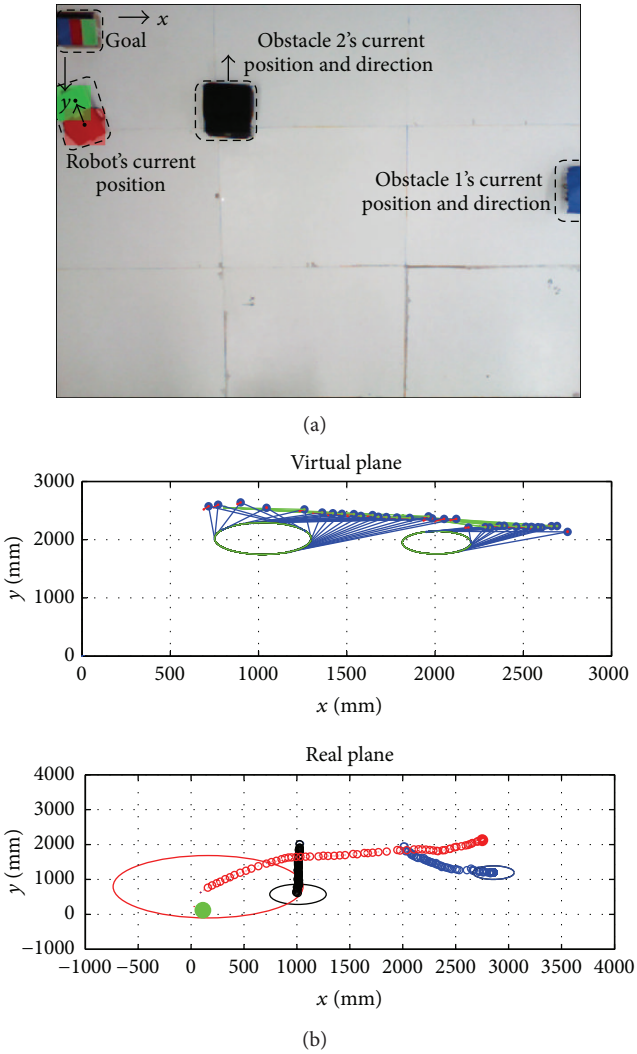


Figure 18: (a) Experiment result after the collision avoidance with all obstacles. (b) Plot of the trajectories of the robot and the obstacles in the virtual and real planes during the collision avoidance with all the moving obstacles.

circles. The smooth trajectory was generated by using the linear navigation laws as explained.

From this experiment, it is easily seen that the proposed hybrid reactive motion planning approach designed by the integration of the virtual plane approach and a sensor based planning is very effective to dynamic collision avoidance problems in cluttered uncertain environments. The effectiveness of the hybrid reactive motion planning method makes its usage very attractive to various dynamic navigation applications of not only mobile robots but also other autonomous vehicles such as flying vehicles and self-driving vehicles.

5. Conclusion

In this paper, we proposed a hybrid reactive motion planning method for an autonomous mobile robot in uncertain dynamic environments. The hybrid reactive motion planning method combined a reactive path planning method which could transform dynamic moving obstacles into stationary ones with a sensor based approach which can provide relative information of moving obstacles and environments. The key features of the proposed method are twofold; the first key feature is the simplification of complex dynamic motion planning problems into stationary ones using the virtual plane approach while the second feature is the robustness of a sensor based motion planning in which the pose estimation of moving obstacles is made by using a Kinect sensor which provides a ranging and color detection. The sensor based approach improves the accuracy and robustness of the reactive motion planning approach by providing the information of the obstacles and environments. The performance of the proposed method was demonstrated through not only simulation studies but also field experiments using multiple moving obstacles.

In the further work a sensor fusion approach which could improve the heading estimation of a robot and the speed estimation of moving objects will be investigated more for more robust motion planning.

Conflict of Interests

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

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) (no. 2014-017630 and no. 2014-063396) and was also supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (no. 2014-066733).

References

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, The MIT Press, London, UK, 2nd edition, 2011.

[2] J. P. van den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, 2005.

[3] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[4] R. Gayle, A. Sud, M. C. Lin, and D. Manocha, "Reactive deformation roadmaps: motion planning of multiple robots in dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3777–3783, San Diego, Calif, USA, November 2007.

[5] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[6] K. P. Valavanis, T. Hebert, R. Kolluru, and N. Tsourveloudis, "Mobile robot navigation in 2-D dynamic environments using

an electrostatic potential field," *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*, vol. 30, no. 2, pp. 187–196, 2000.

[7] X. Gao, Y. Ou, Y. Fu et al., "A novel local path planning method considering both robot posture and path smoothness," in *Proceeding of the IEEE International Conference on Robotics and Biomimetics (ROBIO '13)*, pp. 172–178, Shenzhen, China, December 2013.

[8] J. Jiao, Z. Cao, P. Zhao, X. Liu, and M. Tan, "Bezier curve based path planning for a mobile manipulator in unknown environments," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '13)*, pp. 1864–1868, Shenzhen, China, December 2013.

[9] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3310–3317, May 1994.

[10] G. Song, H. Wang, J. Zhang, and T. Meng, "Automatic docking system for recharging home surveillance robots," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 428–435, 2011.

[11] M. Seder and I. Petrović, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 1986–1991, Roma, Italy, April 2007.

[12] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[13] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.

[14] J. Borenstein and Y. Koren, "The vector field histogram—fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[15] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proceedings of the 13th IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3375–3382, Minneapolis, Minn, USA, April 1996.

[16] J. Minguez and L. Montano, "Nearness Diagram (ND) navigation: collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, 2004.

[17] E. Owen and L. Montano, "Motion planning in dynamic environments using the velocity space," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 997–1002, August 2005.

[18] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: a collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–574, 1998.

[19] F. Belkhouche, T. Jin, and C. H. Sung, "Plane collision course detection between moving objects by transformation of coordinates," in *Proceedings of the 17th IEEE International Conference on Control Applications Part of IEEE Multi-Conference on Systems and Control*, San Antonio, Tex, USA, 2008.

[20] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[21] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[22] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: obstacles moving along arbitrary trajectories," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 3716–3721, Seoul, Republic of Korea, May 2001.

[23] F. Belkhouche, "Reactive path planning in a dynamic environment," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 902–911, 2009.

[24] B. Dorj, D. Tuvshinjargal, K. T. Chong, D. P. Hong, and D. J. Lee, "Multi-sensor fusion based effective obstacle avoidance and path-following technology," *Advanced Science Letters*, vol. 20, pp. 1751–1756, 2014.

[25] R. Philippsen, B. Jensen, and R. Siegwart, "Towards real-time sensor based path planning in highly dynamic environments," in *Autonomous Navigation in Dynamic Environments*, vol. 35 of *Tracts in Advanced Robotics*, pp. 135–148, Springer, Berlin, Germany, 2007.

[26] K.-T. Song and C. C. Chang, "Reactive navigation in dynamic environment using a multisensor predictor," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 6, pp. 870–880, 1999.

[27] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[28] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: a review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[29] F. Kammergruber, A. Ebner, and W. A. Gunthner, "Navigation in virtual reality using microsoft kinect," in *Proceedings of the 12th International Conference on Construction Application of Virtual Reality*, Taipei, Taiwan, November 2012.

[30] Microsoft Kinect for X-BXO 360, http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one.

[31] K. Kronolige and P. Mihelich, "Technical Description of Kinect Calibration," July 2014, http://www.ros.org/wiki/kinect_calibration/technical.

[32] G. Shashank, M. Sharma, and A. Shukla, "Robot navigation using image processing and isolated word recognition," *International Journal of Computer Science and Engineering*, vol. 4, no. 5, pp. 769–778, 2012.

[33] T. W. Choe, H. S. Son, J. H. Song, and J. H. Park, *Intelligent Robot System HBE-RoboCar*, HanBack Electronics, Seoul, Republic of Korea, 2012.

[34] K. R. Jadavm, M. A. Lokhandwala, and A. P. Gharge, "Vision based moving object detection and tracking," in *Proceedings of the National Conference on Recent Trends in Engineering and Technology*, Anand, India, 2011.

[35] A. Chandak, K. Gosavi, S. Giri, S. Agrawal, and P. Kulkarni, "Path planning for mobile robot navigation using image processing," *International Journal of Scientific & Engineering Research*, vol. 4, pp. 1490–1495, 2013.