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

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

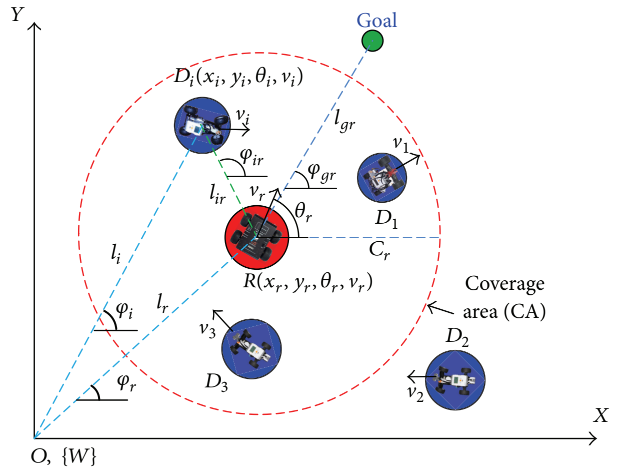


Figure 1: Geometry of the navigation problem. Illustration of the kinematic and geometric variables.

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

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

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:

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

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

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

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

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

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

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

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:

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

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

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

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:

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

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

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

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

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:

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

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

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

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

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

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:

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

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

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

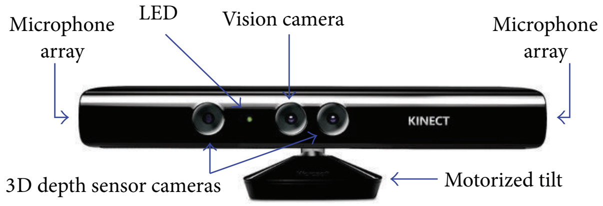


FIGURE 2: Architecture of Microsoft Kinect sensor.

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

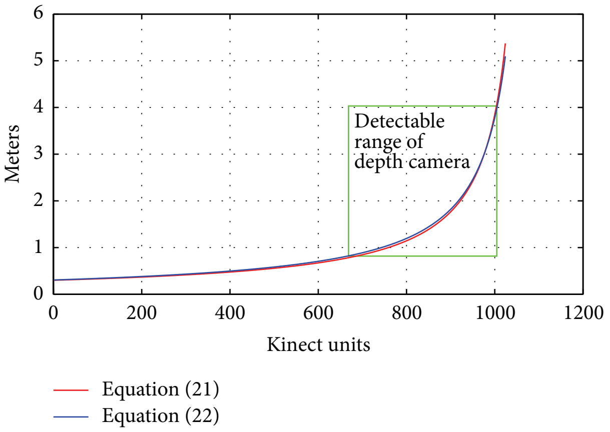


Figure 3: Kinect depth measurement and actual distance.

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

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

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

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.

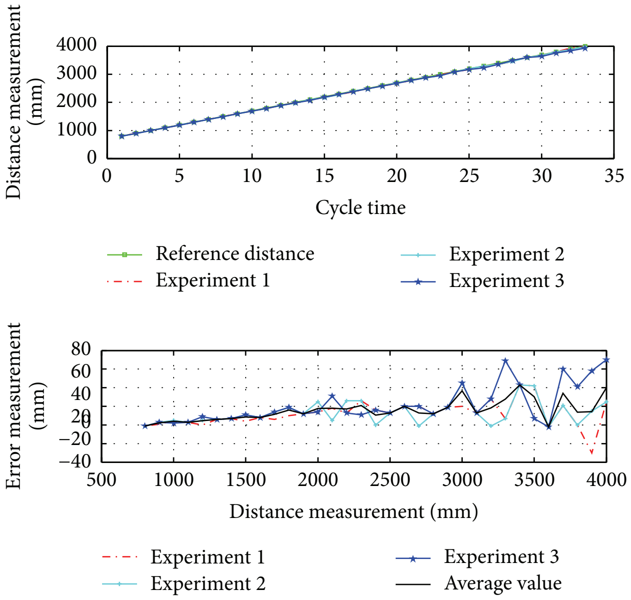


Figure 4: Kinect depth camera measurement experiment and error.

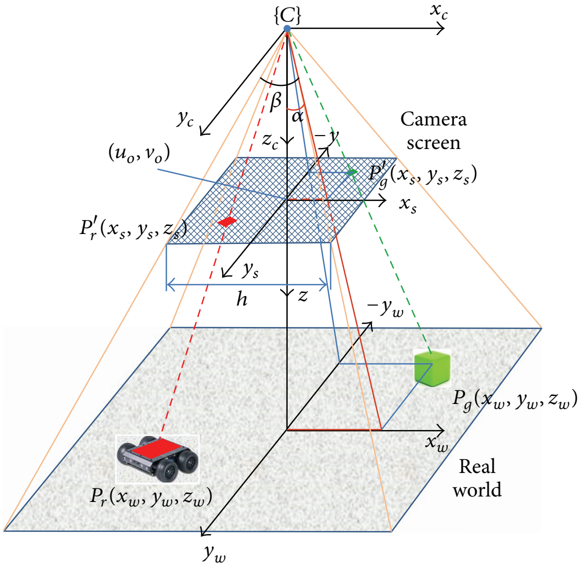
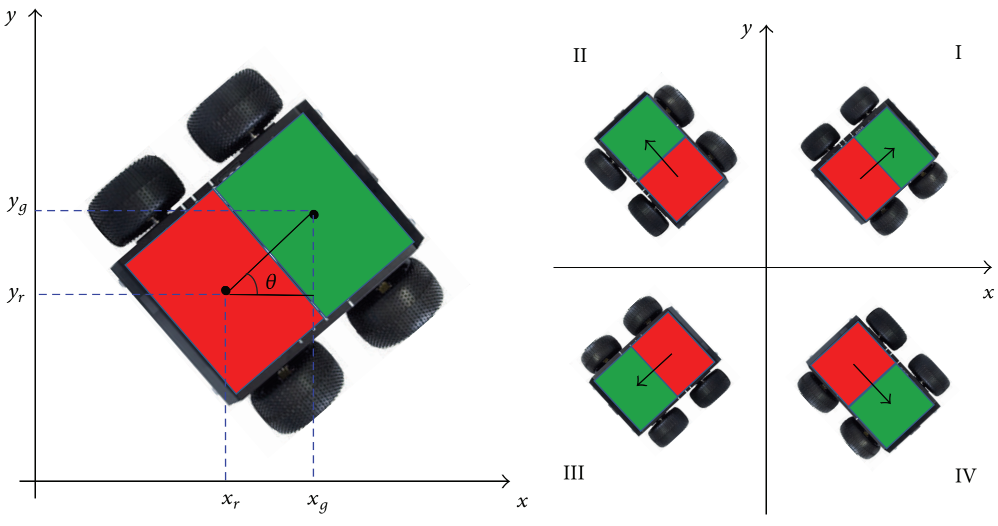


Figure 5: Robot and obstacle localization using the Kinect sensor for ranging and positioning computation.

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

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



(a) (b)

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:

$$\begin{gathered} \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} \dfrac{1}{\rho_u} & 0 & u_0 \\ 0 & \dfrac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} \\ P' = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \dfrac{u'}{w'} \\ \dfrac{v'}{w'} \end{pmatrix}, \qquad \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = C \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}. \end{gathered} \tag{25}$$

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:

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

(2) vertical coordinate:

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

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 $\phi_{gr}$
    Send robot speed
    if All $D_i$ in CA
        Calculate $\dot{l}_{ir}$ using Kinect
        if All $\dot{l}_{ir} > 0$ 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 $\theta_r$ of quick motion control
                    Send robot speed
                else
                    Construct the $\theta$-window
                    Chose the appropriate values for $\theta_r$
                    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:

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

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

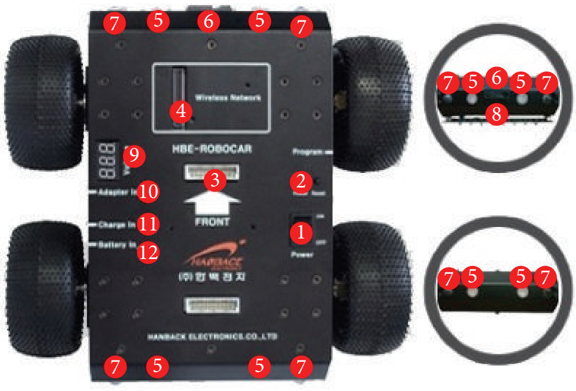
Finally, the relative velocity obstacle based reactive dynamic navigation algorithm with the capability of collision avoidance is summarized in Algorithm 1.

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].



1 Power switch
2 Power LED and reset switch
3 Interface connector
4 Wireless network connector
5 Ultrasonic sensor
6 PSD sensor
7 LED
8 Phototransistors
9 Voltmeter
10 DC in
11 Charge in
12 Battery in

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

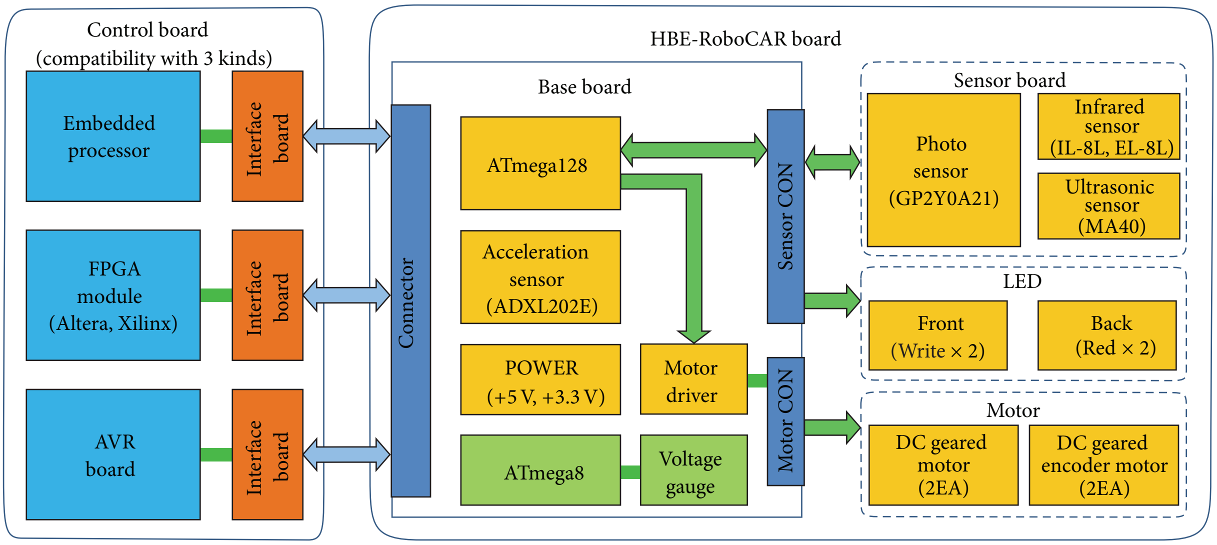


Figure 8: Block diagram of the HBE-RoboCAR embedded system.

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

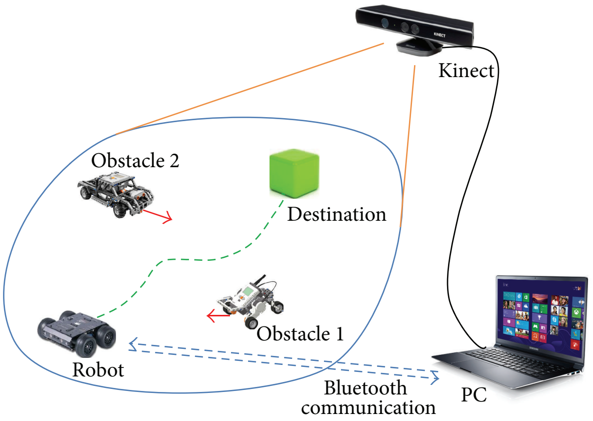
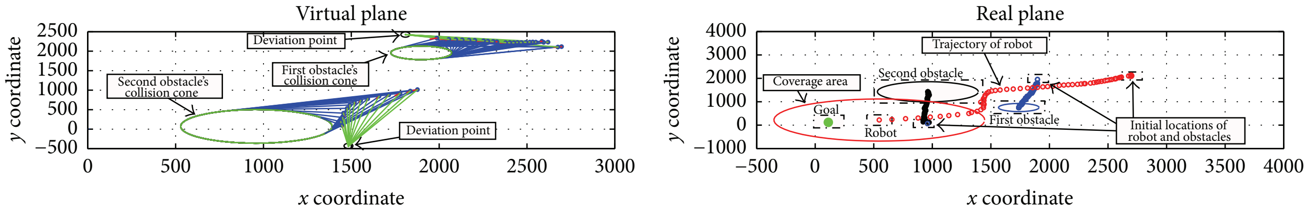


Figure 9: Test scenario and architecture for experimental setup.



(a)

(b)

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

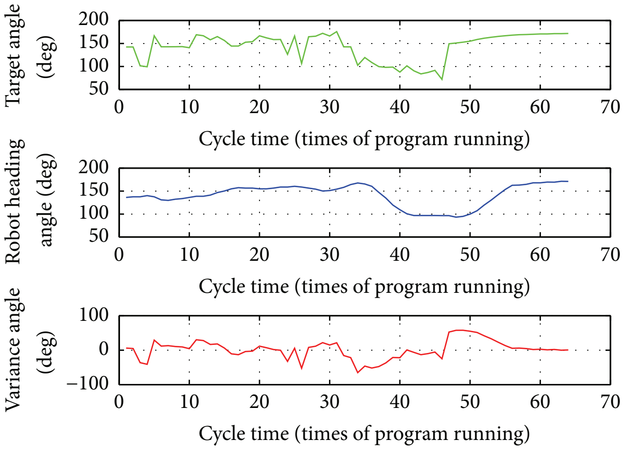


Figure 11: Orientation angle information: target angle, robot heading angle, and variance angle.

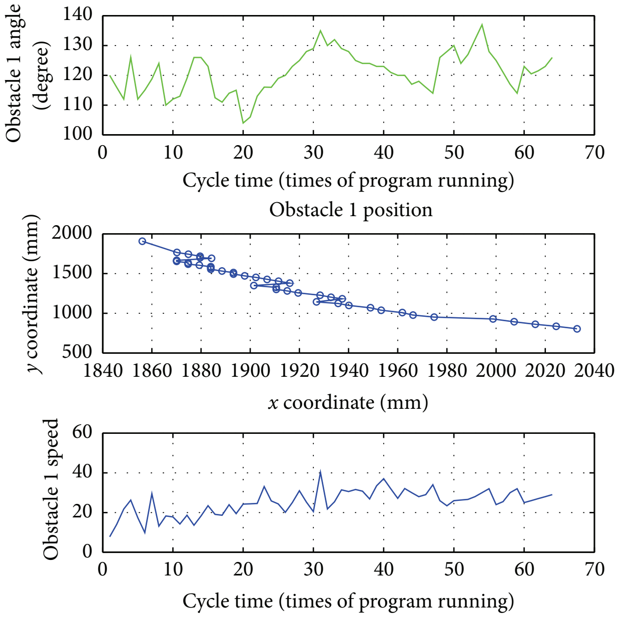


Figure 12: Results of first moving obstacle's orientation angle, speed, and position in x-y coordinates.

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

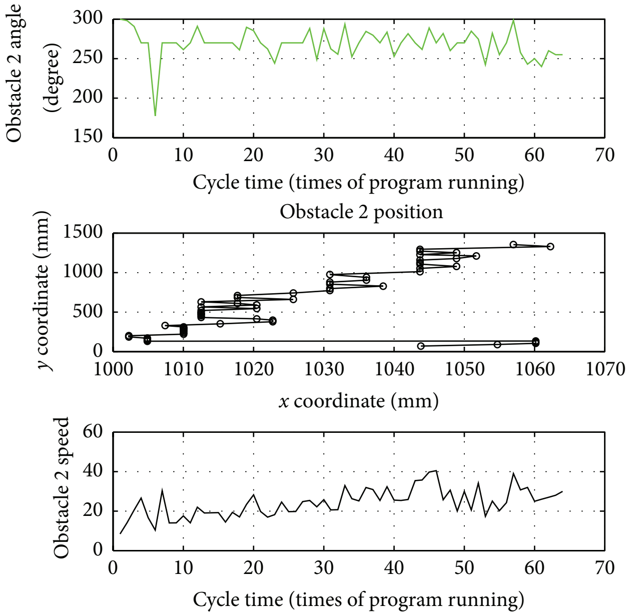


Figure 13: Results of second moving obstacle's orientation angle, speed, and position in x-y coordinates.

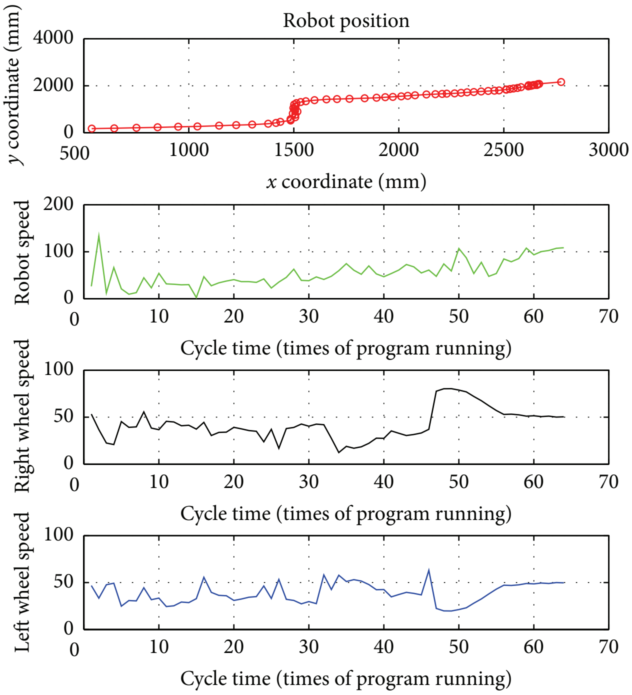
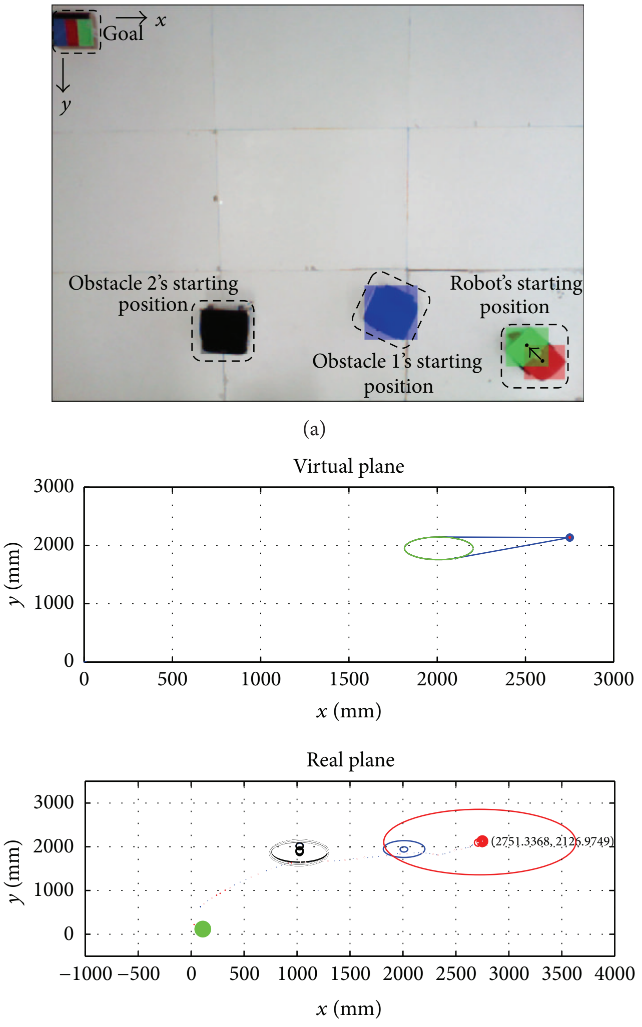


Figure 14: Results of robot's trajectory, speed, and right and left wheels speed.

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.



(a)

(b)

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),

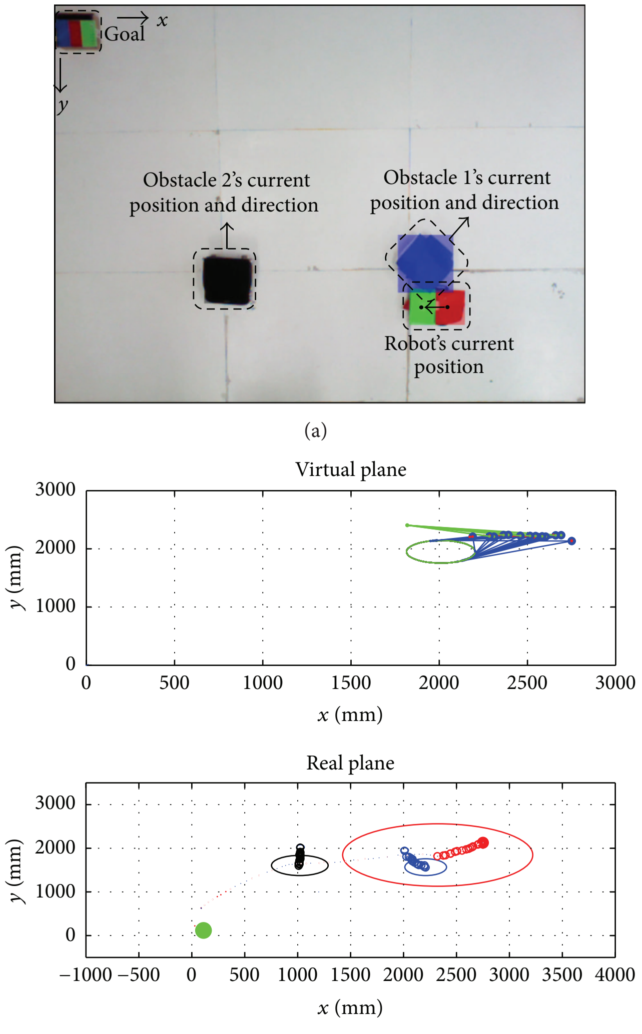


Figure 16: (a) Experiment result of collision avoidance with the first obstacle. (b) Plot of the trajectories of the robot and the obstacles in the virtual and real planes during the collision avoidance with the first moving obstacle.

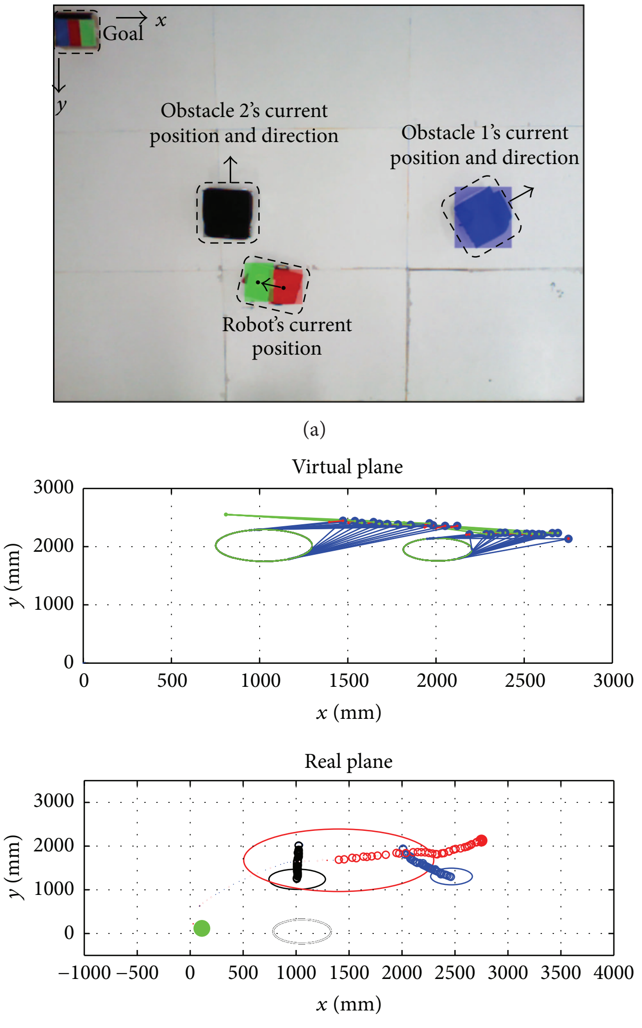


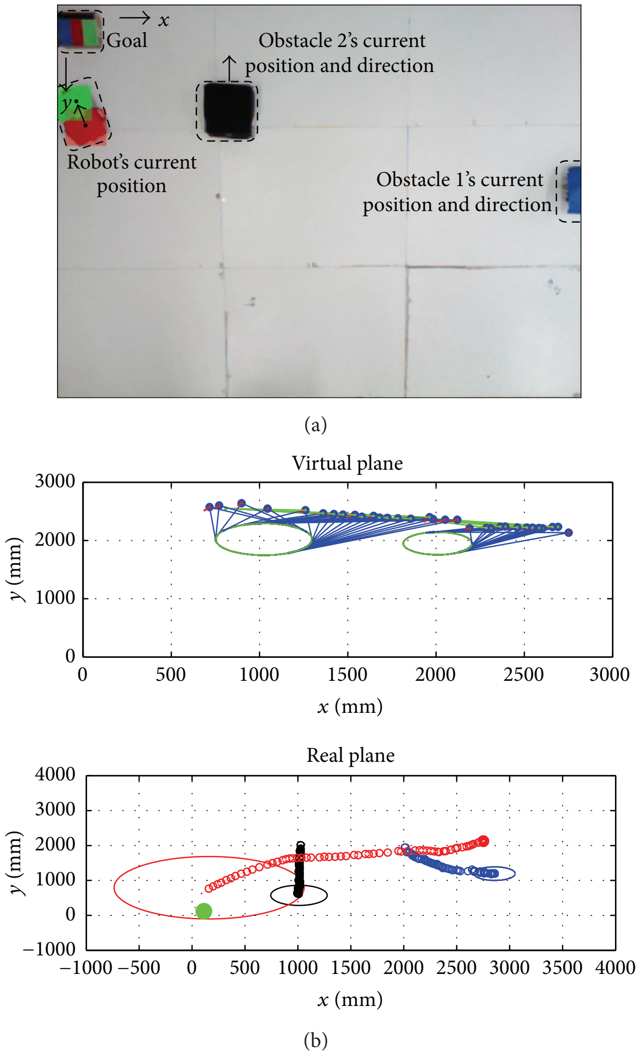
Figure 17: (a) Experiment result of collision avoidance with the second obstacle. (b) Plot of the trajectories of the robot and the obstacles in the virtual and real planes during the collision avoidance with the second moving obstacle.

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



(a)

(b)

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.

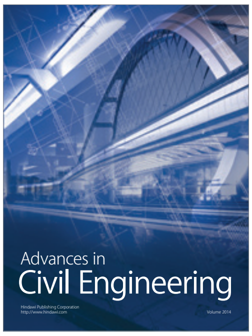
Journal of
Engineering

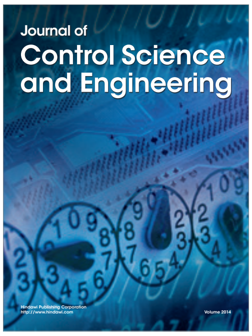
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

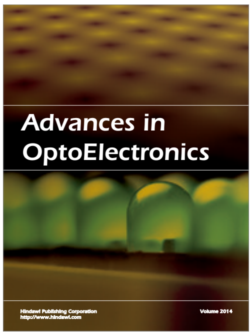
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

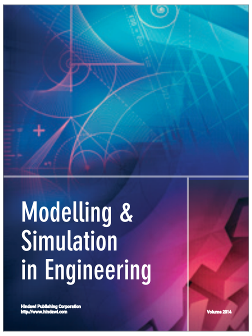
Journal of
Robotics

Journal of
Electrical and Computer Engineering

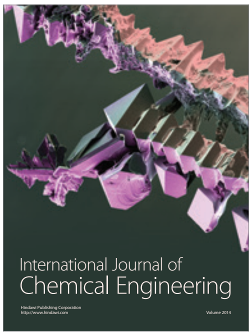

Advances in
OptoElectronics

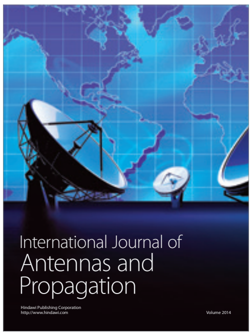
VLSI Design

International Journal of
Navigation and Observation

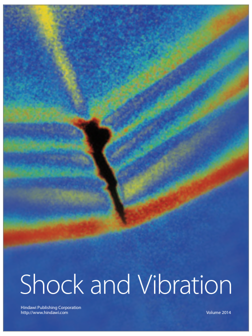

Modelling & Simulation in Engineering

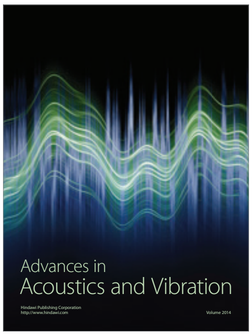
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration