# A New Method of Obstacle Avoidance for Service Robots in Indoor Environments

**Widodo Budiharto[1], Ari Santoso[2], Djoko Purwanto[2]& Achmad Jazidie[2]**

[1]Bina Nusantara University- Jakarta
[2]Electrical Engineering Department, Faculty of Industrial Technology
Institute of Technology Sepuluh Nopember Surabaya (ITS)
Kampus ITS, Keputih, Surabaya 60111, Indonesia
Email: wbudiharto@binus.edu

**Abstract.** The objective of this paper is to propose an obstacle avoidance method for service robots in indoor environments using vision and ultrasonic sensors. The service robot in this research is programmed to deliver a drinking cup to the recognized customer from a specified starting point. We have developed three main modules: one for face recognition, one to identify obstacles, and one to maneuver. The detection of static obstacles, which is based on information from the landmark, uses an edge-detection method and is used to generate a planned path. The speed, direction and distance of the moving obstacle are measured by using vision and distance sensors in order for the robot to make a maneuver. Algorithms for obstacle avoidance for service robots are proposed and a new geometrical model is introduced for service robots to make a good avoidance maneuver. The main aim of this research is to provide a complete mechanism for obstacle avoidance by a vision-based service robot, whereas popular obstacle avoidance methods such as PVM do not provide such a feature. We present the results of an experiment with a service robot in which our proposed method was implemented, after which we evaluate its performance.

**Keywords:** *moving obstacle; obstacles avoidance; robot vision; service robot.*

## 1 Introduction

Service robot development is an emerging technology that will be popular in the future. A typical application is a service robot that can recognize people and obstacles indoors or outdoors and accomplish a specific task given by a user. Obstacle avoidance for service robots in indoor environments under varying illumination conditions is complex and challenging. The service robot has to maneuver in order to avoid static and moving obstacles and reach its target in an optimal manner. This necessitates robust algorithms for collision avoidance, path planning and autonomous action by the service robot [1-4].

In recent years, many techniques have been developed to carry out obstacle avoidance efficiently by using recent sensor data [5,6]. General obstacle

Received September 30th, 2010, Revised April 27th, 2011, 2nd Revision August 1st, 2011, 3rd Revision February 16th, 2012, 4th Revision May 1st, 2012, Accepted for publication May 2nd, 2012.

avoidance can be accomplished with different sensors, such as vision, sonar and laser sensors. The Potential Field Method (PFM) and Virtual Force Histogram (VFH) for obstacle avoidance have gained increased popularity in the field of mobile robots. PFM is based on attractive and repulsive potential fields that guide the robot to reach its target or to avoid obstacles. PFM has inherent limitations, such as no passage between closely spaced obstacles, oscillations in the presence of obstacles and in narrow passages, and robot still stand when attractive and repulsive forces are equally strong. VFH is a fast obstacle avoidance method based on a polar histogram of obstacles that provides directions for safe travel. VFH also has shortcomings: the results are sensitive to thresholds and it requires expensive hardware because it uses a histogram grid world model that has to be updated by rapidly firing 24 sensors during motion [7,8]. In the case of vision-based service robots, PFM and VFH do not provide methods for obstacle avoidance or maneuvering.

Another main challenge for service robot development is to detect moving obstacles accurately in the landmark, especially using a vision sensor (camera). The most notable algorithms for detecting moving objects from a moving platform using only a vision sensor can be grouped into two distinct classes, those using optical flow and those using qualitative estimates of motion. A disadvantage of both methods is the difficulty to compute the optical flow with an acceptably low level of noise [9], and the moving entities must be identified and their future position needs to be predicted over a finite time [10-13].

A literature study by the authors shows that in many research projects, such as [1] and [2], the task of the service robot is the setting and clearing of tables in a controlled environment. However, no flexible obstacle avoidance method for vision-based service robots in indoor environments is available. In previous papers, we have already proposed a novel method for static and moving obstacle avoidance for vision-based service robots using Bayesian Filtering and the ANFIS Controller, but this method uses predefined maneuvering [14,15]. Obstacle avoidance methods based only on ultrasonic sensors must account for the sensors’ shortcomings, such as inaccuracies, crosstalk, and spurious readings, therefore we have considered a combination of vision and ultrasonic sensors. Vision and ultrasonic sensors are important for the face recognition system, the tracking system and distance measurement in vision-based service robots. This paper introduces a new method for obstacle avoidance for service robots in indoor environments using one single camera. A complete mechanism is proposed for vision-based robots which should be able to avoid obstacles autonomously; this is presented in section 2 and 3. Finally, the results of the implementation of the proposed method and algorithms in a service robot, Srikandi II, are presented in section 4. The discussion of the results in section 5 includes a comparison with the PFM and VFH method, and section 6 contains the conclusion.

# 2 Introduction to Vision-Based Service Robots

## 2.1 Principles and Kinematics of Vision-Based Service Robots

The robot used in this research is a mobile robot which is equipped with two actuator wheels and is considered as a system subject to non holonomic constraints. Consider a basic configuration consisting of an autonomous wheeled mobile robot, its target and a moving obstacle, as shown in Figure 1. When a robot inside an indoor environment moves from its start position to its target, there is the possibility of a moving obstacle hitting the robot. The robot needs relevant information, such as the distance of the obstacle $d_O$, the velocity of the obstacle $v_O$, the orientation of the robot $\theta_R$, and the velocity of the robot $v_R$, in order to be able to take action to avoid the obstacle and make a maneuver with a certain direction angle.

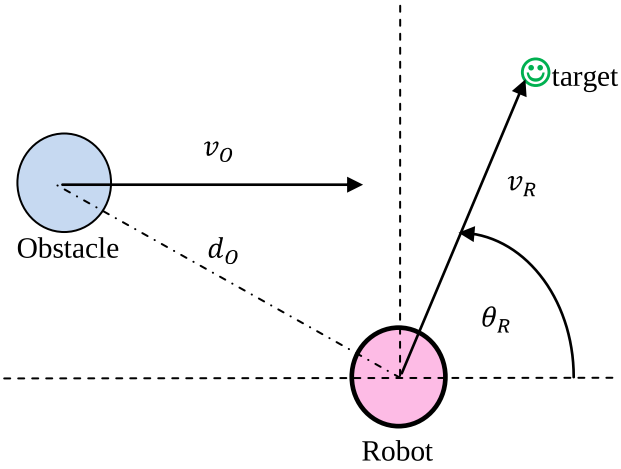


**Figure 1** Configuration of mobile robot, moving obstacle and target.

Based on the configuration in Figure 1, we propose the model of a vision-based service robot using a single camera and a moving obstacle as shown in Figure 2. The camera is an important sensor if we want the robot to identify specific objects (such as a face, a small object, a shape, etc.), which cannot be identified by other sensors, such as an ultrasonic sensor. A camera as a vision sensor has view angle limitations in capturing an object, so we define $\theta_{Cam}$ as the maximum angle at which the moving obstacle can be detected by the camera used in this research.

We identify $\theta_{O,}\theta_{OR,}\theta_{Cam},\theta_R,v_R$ and $v_O$ as very important properties to calculate whether a robot will collide with the moving obstacle or not. Based on Figure 2, we define the angle between the moving obstacle and the robot $\theta_{OR}$ and the orientation of the robot $\theta_R$ as:

$$\theta_{OR} = 180^{\circ} - (\theta_R + \theta_{cam}) \tag{1}$$

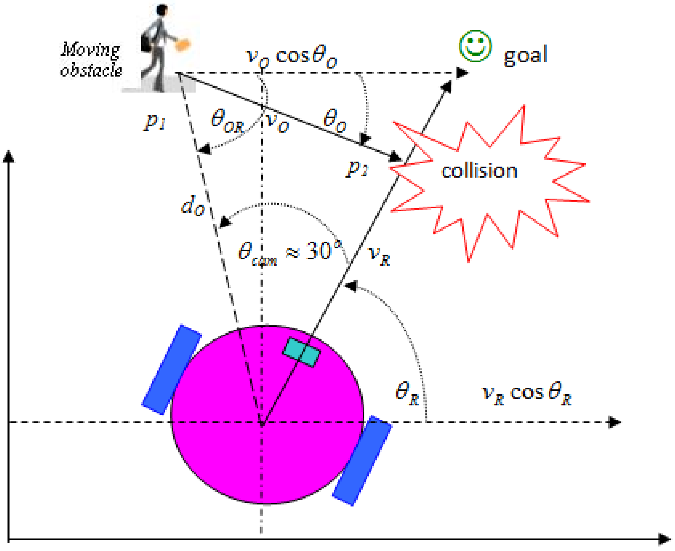


**Figure 2** Proposed Cartesian representation of robot using single camera, moving obstacle and target.

Theoretically, to measure the speed of an obstacle, robots should continuously track the obstacle using a vision sensor such as a camera. We propose the model to calculate the $v_O$ that moves with an angle $\theta_O$ detected by the camera, while at the same time the robot moves with the speed $v_R$ to the target with an angle $\theta_R$. In our method we need 2 points of tracked images ( $p_1$ and $p_2$) with an interval time $t$, after which the difference in pixel position is obtained. Based on Figure 2, the equation to estimate $v_O$ when an obstacle and a robot are moving, and the obstacle appears from the left side of the robot is:

$$v_O \cos\theta_O = \frac{|\,p_2 - p_1\,|\,s}{t} + v_R \cos\theta_R \tag{2}$$

Finally, we can simplify the Eq. (2) and add constraints, such as:

$$v_O = \begin{cases} \dfrac{|p_2 - p_1|s}{t\cos\theta_O} + \dfrac{v_R\cos\theta_R}{\cos\theta_O} & \text{if } 0° <= \theta_O <= 60° \\ v_O \max \quad \text{if } 60° < \theta_O <= 90° \end{cases} \quad (3)$$

From Eq. (3), if $\theta_O$ between 60°< $\theta_O$ <=90°, it is difficult to estimate the velocity of the obstacle, especially when $\theta_O$ =90° (direct approach of the robot), therefore in this situation we assume $v_O$ at the maximum velocity allowed (we define $v_O$ max =80 cm/s). If the calculated result of $v_O$ >=80 cm/s, then we set $v_O$ as the maximum velocity allowed. $p_1$ and $p_2$ are the positions of the moving obstacle in pixels and the scaling factor is in cm/pixel. In principle, equation (3) can also be used for a moving obstacle that appears from the right side of the robot. We also propose a mechanism to predict a collision, using time *t*. A collision occurs when the robot hits the person as a moving obstacle that moves with a specific orientation, as shown in Figure 2, while we assume that the moving obstacle has the intelligence not to hit the robot when its stands still. T is used as a time threshold, and can be calculated using the following formula:

$$t = \frac{d_O \sin\theta_{OR}}{(v_R \sin\theta_R + v_O \sin\theta_O)} \quad (4)$$

where: if *t*<= *T* then the robot stops

if *t*>*T* then the robot moves forward

## 2.2 Architecture of Vision-Based Service Robots

Our system consists of a single camera which obtains a frontal view for object tracking and for the face recognition system and static obstacle detection, a compass and 3 ultrasonic sensors for distance measurement. Figure 3 shown below is a prototype of Srikandi II.

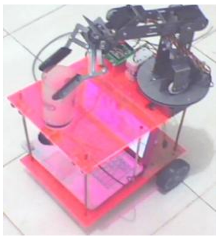


**Figure 3** A vision-based service robot called Srikandi II, developed and used in this research, equipped with a 4 DOF arm robot. The camera has a 640x480 pixel

resolution. A compass sensor and 3 ultrasonic sensors are used as additional sensors for distance measurement, and a laptop is used for image processing[15].

There is an interface program on the laptop for coordinating the robot controller. A 4 DOF arm robot with predefined motion is used to hand over a cup to the customer when the robot arrives at the target position. 1 master controller is used as coordinating actuator and for communication with the laptop, and another controller is used for distance measurement. Figure 4 is an architecture for service robot Srikandi II, utilizing a single camera, a compass and 3 distance sensors. Because this robot needs to recognize and track people, many integrated supporting functions were developed, such as a face recognition system, static and moving obstacles detection, and moving obstacle tracking. We developed a framework for an efficient faces database used by the face recognition system for recognizing the customer.

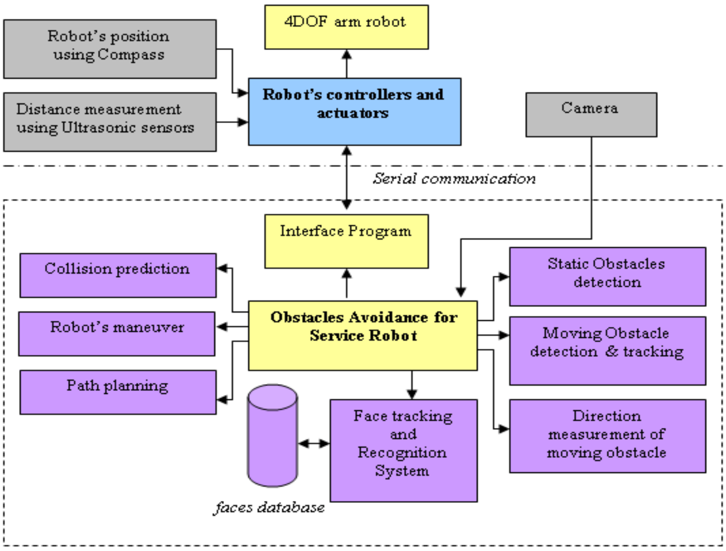

Robot's position using Compass
Distance measurement using Ultrasonic sensors
4DOF arm robot
Robot's controllers and actuators
Camera
Serial communication
Interface Program
Collision prediction
Robot's maneuver
Path planning
Obstacles Avoidance for Service Robot
Static Obstacles detection
Moving Obstacle detection & tracking
Direction measurement of moving obstacle
Face tracking and Recognition System
faces database

**Figure 4** General architecture of the service robot Srikandi II. Hardware and software parts are separated by the dashed line. All arrows indicate dataflow.

# 3 Proposed Method

## 3.1 General Obstacles Avoidance Method

The robot needs information from the ultrasonic sensors about the distance between the obstacle and the robot to be able to avoid a collision,. Ultrasonic sensors generally obtain the radial measurement of the nearest distance from the object located in the area A, as shown in Figure 5. Ultrasonic sensors work at a frequency of 40 KHz and have a deviation angle maximum of about 30°, so usually robots need more than one sensor to be able to measure the distance of an obstacle in its vicinity. The main weakness of this type of sensor is the interference between different sensors and the limited ability to identify the obstacle. The advantage of this type of sensor is that it is usually able to detect the obstacle at a distance >=3 cm, something a vision sensor is not able to do.

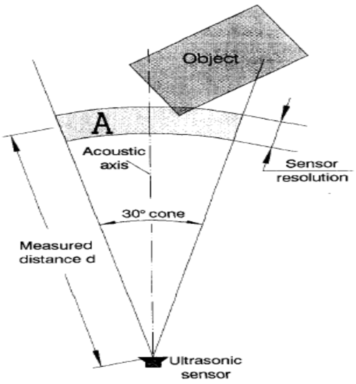


**Figure 5** 2 dimension projection from conical fields of ultrasonic sensor. The distance measurement $d$ indicates the existence of an object in the area [7].

Theoretically, using 3 ultrasonic sensors is enough to obtain distance information from the front, left and right side of the robot. Therefore, we propose the model for obstacle avoidance for a service robot as shown in the flowchart from Figure 6. This method is a combination of static and moving obstacle detection, using vision and ultrasonic sensors. Information about static obstacles is obtained when the robot starts moving.

Images from the landmark are captured and stored in a 640x480 JPEG file and the program will scan the images to find the free area that can be used by the robot. At the end of the program, the path from start to target position is obtained in order to guide the robot in its movement. The next step is object detection and face recognition in order to identify the customer based on PCA (Principal Component Analysis). The information about the moving obstacle is obtained when the robot detects and tracks someone who walks in front of the robot; the distance of the moving obstacle is obtained using distance sensors periodically, let’s say every 1 second. On the basis of the information about the moving obstacle obtained by the vision sensor, we can estimate the speed of the moving obstacle, predict the collision point and determine the maneuver action. Figure 6 shown below is a flowchart that describes the general mechanism for our method to detect and identify an obstacle and make a maneuver to avoid collision:

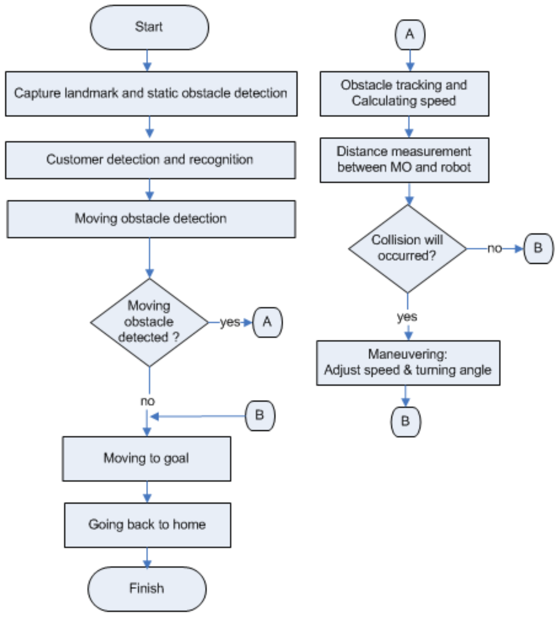

Start
Capture landmark and static obstacle detection
Customer detection and recognition
Moving obstacle detection
Moving obstacle detected ?
yes
A
no
B
Moving to goal
Going back to home
Finish
A
Obstacle tracking and Calculating speed
Distance measurement between MO and robot
Collision will occurred?
no
B
yes
Maneuvering: Adjust speed & turning angle
B

**Figure 6** Flow chart of obstacle avoidance and action from start to target position for a vision-based service robot.

To implement the above flowchart in a service robot that should recognize its customer and have the ability to avoid an obstacles, we have developed algorithms and programs consisting of 3 main modules, namely the face recognition system, the obstacle detection system and the maneuvering method. The algorithm for the general method of obstacle avoidance of a vision-based service robot that moves from a specified start point to a specified target position is as follows:

**Algorithm 1.** General method for obstacle avoidance by a service robot

```
Capture and saving landmark
Call staticObstacleDetection  // using Canny edge detector
Call faceDetectionRecognition  // Face recognition system
if (customer_identified ==true)
    do while not goal // while min_distance < distance measured by front sensor.
        set heading robot to the goal
        call movingObstacleDetectionTracking
        distance measurment
        robot running with normal speed
        if (moving obstacles==true) then
            calculate speed and Distance measurement of moving obstacle
          // if  minimum distance to maneuver reached , then maneuvering
            If ( $d_{safe}$ ==true )
                call maneuveringRobot
    endif
```

```
                move to goal position
            **else**
             Move to goal position  // if moving obstacle==false
           **endif**
        **end while**
        giving a cup to a customer
        move to home / start position // task finished
   **else**
          robot stop // if no customer detected
  **endif**
 function faceRecognition //More detailed in algorithm 2
          face detection and capture face customer
          training images
          testing face customer
          face recognition
 end function
function staticObstacleDetection
          captureLandmark and
          staticObstacleProcessing //More detailed in algorithm 3
 end function
 function movingObstacleDetectionTracking //More detailed in algorithm 3
           Moving obstacle detection
           Moving obstacle tracking
 End function
 function maneuveringRobot //More detailed in algorithm 4.
          calculate $\theta_m$ and $v_R$
          adjust turning angle and speed
          Update position of robot to the goal position
 end function
```

## 3.2 Algorithm for Face Recognition System

We use PCA for the face recognition system to recognize the customer’s face. Illumination and pose variations are mostly responsible for dramatic changes in the appearance of the face, which has proved to be a very difficult problem [16,17]. The brightness level of the face images in an environment is a random variable. For solving the problems related to pose and illumination effects in indoor environments, we have proposed a method for training the system by varying the pose and illumination from the training images using normal distribution generated by our program developed in C++ using the Technical Report Features Pack. Histogram Equalization applied to the input images is a powerful method for automatically standardizing the brightness and contrast of facial images. We have developed a faces database called ITS database, comparing the success rate performance using the Indian database [18] and the Yale University database [19]. The algorithm below shows the face recognition system for our service robot; assumed is that training images are available beforehand; testing images are obtained when the robot detects a face. When a customer is identified, the name of the customer is assigned to the variable *customer_name*.

**Algorithm 2**. Face recognition system for a service robot.

```
// Function to store images as a faces database and train the faces.
Function captureAndTrainingImages
   capture face customer 640x480 pixel using 3 pose (front, left, right)
   do histogram equalization to face customer
    save to database
   training face images
end function

// Function to detects and recognizes a customer
 function faceDetectionRecognition
              face detection                 // Using Haar cascade classifier
           if  (face_detected ==true) then
            capture face customer 640x480 pixel
               face recognition          // testing face customer to faces database
                if( (face_recognition==true) then
                     customer_identified=true
                     set  customer_name
               else
                   customer_identified=false; // no action for the robot if customer not identified
            endif
  end function
```

## 3.3 Algorithm for Obstacle Detection, Obstacle Tracking and Path Planning

The algorithm for detecting fast static obstacles uses the Canny edge detector, moving obstacle tracking and path planning as shown in algorithm 3 below; input landmark: 640x480 pixel in JPEG format. The calibration is needed for getting the threshold value for setting the pixel to white/black. A white area indicates a free area. After that, the target position is obtained based on the most free area. The predefined start and target position are used for path planning by the robot. Finally, the program will guide the robot to the target position using the compass sensor. When a moving obstacle is detected, it will be tracked and an estimation of the direction angle $\theta_O$ and velocity of the obstacle $v_O$ will be calculated.

**Algorithm 3**. Fast obstacle detection and path planning.

```
// Function to detect static obstacle using edge detection and path planning
function staticObstacleDetection
Capture and saving landmark
Canny edge detection and closing operation
while (not end of pixel from image(x,y) )
   if(avg. pixel >threshold)
               set pixel to  white
   else
        Set pixel to black
  endif
   end while

     Smoothing operation
     Display start and goal position  // draw a track from start to goal position
     Read position by compass sensor  // for heading the robot to goal position
     Set heading robot to goal position
 end function

// Function to detects and tracks a moving obstacle
 function movingObstacleDetectionTracking
            moving obstacle detection // Using Haar cascade classifier
               if (moving_obstacle==true) then
                      distance measurement between  robot and moving obstacle
                      //  tracking moving obstacle using motion history
                            different=frame_pre –frame_now // different of motion
                            Update motion history
                             Calculate motion gradient
                            $\theta_O$ = 360-Globalorientation(direction) // $\theta_O$ is obtained
                           Calculate $v_O$
              endif
 end function
```

### 3.4 Method and Algorithm for Maneuvering the Robot

In this paper, we introduce a new geometrical model for a robot to maneuver in order to avoid a moving obstacle. 3 ultrasonic sensors, at the front, left and right side of the robot, continuously measure the distance between the robot and a moving obstacle to ensure that the distance between the robot and the obstacle remains larger than the minimum distance to avoid collision. The left and right sensors also make sure that if the moving obstacle comes from the left or right side of the robot it will be identified by these sensors. The proposed model to make a maneuver is based on the model shown in Figure 7.

From Figure 7, in which the constant of the collision angle is denoted as $k_\theta$ and the minimum distance to start maneuvering as $d_{safe}$: when the robot detects that the minimum distance to start maneuvering is reached, in order to avoid collision it will use the turning angle $\theta_m$ if robot is at the right/left side of the moving obstacle:

$$\theta_m = \begin{cases} \theta_R + (\frac{\pi}{2}(k_\theta / d_O)) \text{ if } \mathrm{d_O < d_{safe}}, \text{robot at the right side of obs.} \\ \\ \theta_R - (\frac{\pi}{2}(k_\theta / d_O)) \text{ if } d_O < d_{safe}, \text{robot at the left side of obs.} \end{cases} \tag{5}$$

If $d_n$, the minimum distance to reorientate the robot to the target position, is reached, the turning angle is:

$$\theta_m = \theta_R \text{ if } d_{safe} > d_O > d_n \tag{6}$$

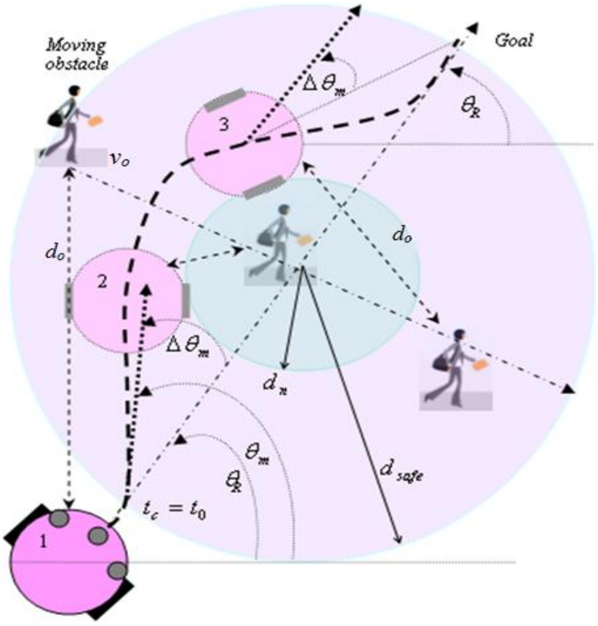


**Figure 7** Model of a maneuvering method to avoid a moving obstacle. This shows a robot maneuvering smoothly to avoid collision with a moving object based on a calculation of the distance between the robot and the moving obstacle.

If $d_{safe} > d_O > d_n$ is reached, the robot will start to reorientate to the target using its compass sensor, based on previously obtained data. The velocity of the robot $v_R$ when in avoidance mode is computed between $\frac{v_{\max}}{2}$ - $v_{\max}$; the maximum speed occurs when $\Delta\theta_m$ is $\frac{\pi}{2}$, for whichthe proposed formula is:

$$v_R = v_{\max}\left(1 - \left|0.5 - \frac{\Delta\theta_m}{\pi}\right|\right) \tag{7}$$

Based on the geometrical model shown in Figure 6, the algorithm for the maneuvering method is:

**Algorithm 4.** Maneuvering method for robot.

```
// Function to maneuver the robot to avoid collision with a moving obstacle
function maneuveringRobot
 if $d_O < d_{safe}$ then
```

```
        set maneuver=true
        calculate $\theta_m$ and $v_R$ for maneuver
        adjust turning angle and speed
endif
if $d_{safe} > d_O > d_n$ then
        set maneuver==false
        set $\theta_m = v_R$ for reorientation to the goal
        adjust turning angle and speed
endif
        Update position of robot to the goal position
end function
```

## 4 Experimental Results

For conducting experiments with obstacle avoidance, a self-navigating service robot has been programmed to deliver a cup in our 4x5 meters lab as a simulation of an indoor environment. Face tracking and recognition based on eigenspaces with 3 images per person were used and databases for the images have been developed. Because of the limitations of the space, people as moving obstacles only walked straight in front of the robot and approached the robot directly. The normal speed of the robot was 0.2 m/s and varied when the robot maneuvered. We defined *min_distance* (between robot and obstacle) as 40 cm in order for the robot not to collide with the obstacle as an emergency response. We set up the experiment in such a way that a moving obstacle would collide with the robot, using the parameters shown in Table 1:

**Table 1** Variables used in our experiments.

| No | Variables | Values |
|---|---|---|
| 1 | $\theta_O$ | $0^\circ$-$80^\circ$ |
| 2 | $v_O$ | 0.2-0.8 m/s |
| 3 | $v_R$ | 0.2-0.4m/s |
| 4 | $T$ | 1s |
| 5 | $\theta_{Cam}$ | $30^\circ$ |
| 6 | $d_{safe}$ | 120 cm |
| 7 | $d_n$ | 80cm |
| 8 | $k_\theta$ | 60 |
| 9 | min_distance | 40 cm |

Figure 8 shown below is the set-up of an experiment carried out for this research. We used a chair and a table as static obstacles and someone who walked straight infront of the robot and approached the robot as a moving obstacle.

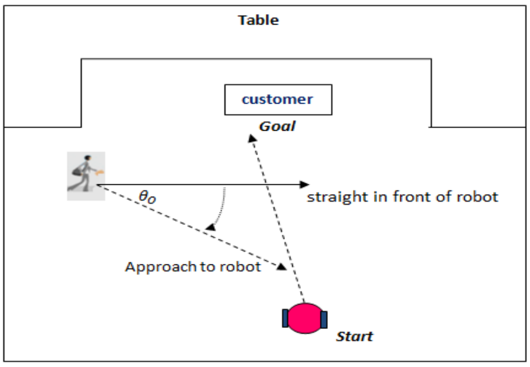
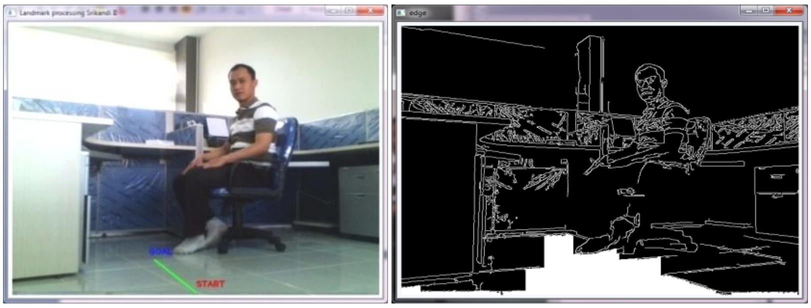


**Figure 8** Set-up experiment for detecting static and moving obstacles, navigating and maneuvering robot from start to target.

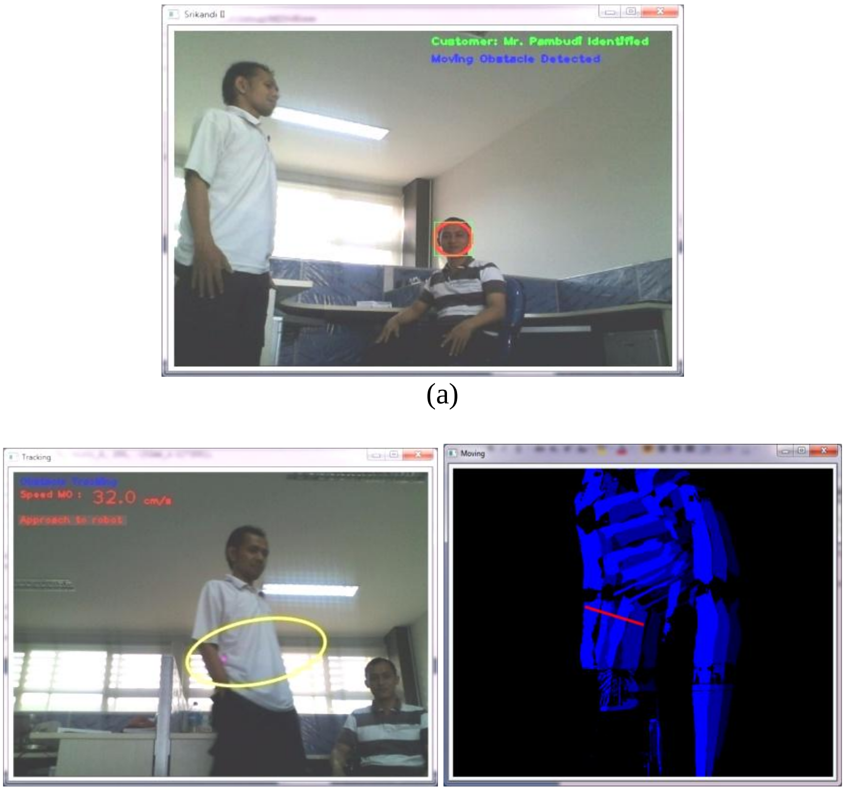
The procedure of the experiment was as follows. First, the robot will capture the landmark and process the images to determine static obstacles and the start-target position. Then the robot will detect if the customer is there or not; if the customer is there, the program will detect and track moving obstacles. Figure 9 shown below indicates that our method for detecting static obstacles worked well in our experiment and our program was able to determine the free path from the start position to the target position. We used the Canny edge detector's closing operation and smoothing to reduce noise from the images. We used the upper body feature for moving obstacle detection and the frontal face feature from OpenCV library [20].



(a) (b)

**Figure 9** Result of simulation for service robot for detecting static obstacles and path planning from start to target position. Straight line from start to goal as a guidance for the robot to move (a). Edge detecting and image processing for finding static obstacles. The white area indicates space with no obstacles (b).

The next step is customer detection and recognition. We evaluated the result of our proposed face recognition and compared it with the Indian and Yale University faces databases. During our experiment, the ITS and Yale University faces databases had a higher success rate than the Indian faces database when we varied or did not vary the illumination [18]. When a moving obstacle appeared and collision threatened, the robot would maneuver to avoid the obstacle, while the turning angle and speed were based on eq. 5-7. For establishing the direction of the moving obstacle, we use a tracking and motion history method as shown in Figure 10.



Customer: Mr. Pambudi Identified
Moving Obstacle Detected

(a)

Speed MO : 32.0 cm/s
Approach to robot

(a)

(b)

**Figure 10** An improved Face Recognition System to identify a customer (a). Detecting and tracking a moving obstacle and information about the speed of a moving obstacle displayed (b). Using motion history with the direction of the movement along the red line indicating that the person approaches the robot (c). Upper body library is used for detecting moving obstacles in this program.

Figure 11 shown below is the general result of the experiment with obstacle avoidance by a service robot. It shows clearly that the robot avoided the obstacle and moved to the target safely with the proposed maneuvering method.

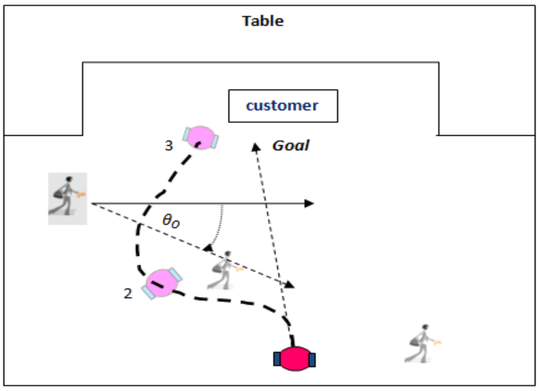
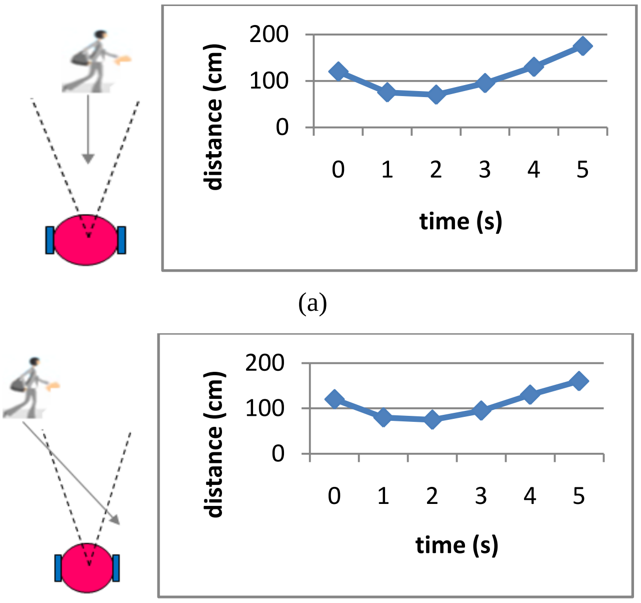


**Figure 11** Result of experiment with obstacle avoidance by a service robot in an indoor environment. It shows that the robot successfully avoided the static and moving obstacles while reaching the target position.

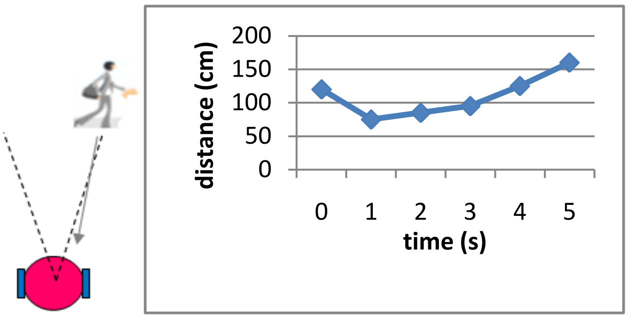
To identify the behavior of obstacle avoidance using the proposed method, we examine our method on the basis of some quantitative measurements for different scenarios. We want to know the behavior of our method if the direction of the moving obstacle is varied; see Figure 12.



(a)

(b)

**Figure 12** Result of experiment using different scenarios (a), (b) and (c) to obtain the behavior of obstacle avoidance using our method. The speed of the obstacle $v_o$ is 40 cm/s and the speed of the robot $v_R$ is 20 cm/s.



(c)

**Figure 12** ***Continuous***. Result of experiment using different scenarios (a), (b) and (c) to obtain the behavior of obstacle avoidance using our method. The speed of the obstacle $v_o$ is 40 cm/s and the speed of the robot $v_R$ is 20 cm/s.

## 5 Results Discussion

In general, the proposed method for obstacle avoidance has been successfully implemented and it has shown a good performance. 3 ultrasonic sensors succeeded to detect and measure the distance of a moving obstacle continuously. Noise or imperfect distance measurement still gave the expected action for the robot to maneuver smoothly. The adjustments of the angle of maneuver and the speed of the robot were very smooth because they only depend on the distance of the moving obstacle proportionally, according to the proposed formula.

There is no oscillation when the robot travels to the target because the robot uses path planning and is guided by a compass. The robot is able to identify a customer because the face recognition system has enough training images to adapt to pose and illumination variations. An indication of the speed of performance of our method and algorithms is the sampling time $S$ (i.e. the speed at which the steer and speed commands by the low-cost controller are issued. The following events occur during $S$ :

1. Obtain sonar information from the sensor controller
2. Calculate the moving obstacle distance
3. Process the information about the moving obstacle from the vision sensor
4. Calculate the speed command and determine the action
5. Communicate with the motion controller in order to send speed and steer command.

On an Intel Atom 1.6 GHz and 1 GB RAM laptop $S$ =220 ms. This makes the proposed system reliable and fast for obstacle avoidance by service robots.

Table 2 provides a comparison between our results and common methods PFM and VFH to prove that our method has the important additional feature that it can be used with low-cost sensors:

**Table 2** Comparison between PFM, VFH and our method.

| No | Variables | PFM | VFH | Our method |
|---|---|---|---|---|
| 1 | Provide a complete mechanism for a vision-based service robot | x | x | √ |
| 2 | Low-cost sensors | √ | x | √ |
| 3 | Fast obstacle avoidance | √ | √ | √ |
| 4 | No oscillations in the presence of obstacles | x | √ | √ |

## 6 Conclusions

This paper presents a new method of obstacle avoidance for service robots in indoor environments. Algorithms for obstacle avoidance and a geometrical model for making a maneuver were implemented in a service robot. Integration of vision and ultrasonic sensors proved applicable as main sensors for a service robot. Experimental results with various scenarios have shown that the robot reaches the target point while avoiding moving obstacles with the proposed method, and is not limited by an increase in the complexity of the system as the environment changes. The obstacle avoidance proposed has shown a good performance and could be an important feature especially of a vision-based service robot. There is no oscillation when the robot travels to the target because the robot uses path planning and is guided by a compass. The sensor system is very cheap because it only uses 3 distance sensors. From the quantitative measurements it follows that the obstacle avoidance based on our method worked very well for different scenarios. In the future, we will model a navigation system for a vision-based service robot with the ability to identify multiple moving obstacles and additional state estimation using the Bayesian approach.

## References

[1] Acosta, L., González, E.J., Rodríguez, J.N., Hamilton, A.F., Méndez J.A., Hernéndez S., Sigut S.M. &Marichal G.N., *Design and Implementation of a Service Robot for A Restaurant*, International Journal of robotics and automation, **21**(4), pp. 273-281, 2006.

[2] Qing-wiau, Y., Can, Y., Zhuang, F. & Yan-Zheng, Z., *Research of the Localization of Restaurant Service Robot*, International Journal of Advanced Robotic Systems, **7**(3), pp. 227-238, 2010.

[3] Borenstein, J. & Koren, Y., *The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots*, in proc. IEEE Trans. On Robotics and Automation, **7** (3), pp.278-288, 1991.

[4] Kahraman, F., Kurt, B.& Gokmen, M., *Robust Face Alignment for Illumination and Pose Invariant Face Recognition*, In-Tech publishing, pp.239-240, 2010.

[5] Masehian, E.& Katebi, Y., *Robot Motion Planning in Dynamic Environments with Moving Obstacles and Target*, Int. Journal of Mechanical Systems Science and Engineering, **1**(1), pp. 20-25, 2007.

[6] Minura, J., Uozumin, H. & Shirai, Y., *Mobile Robot Motion Planning considering the Motion Uncertainty of Moving Obstacles*, in Proc. IEEE Int. Conf. on System, Man, and Cybernetics, pp. 692-698, 1999.

[7] Khatib, O., *Real-time Obstacle Avoidance for Manipulator and Mobile Robots*, International Journal of Robotics Research **5**(1), pp.90-98, 1986.

[8] Borenstein, J. & Koren, Y., *Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation*, in proc. IEEE Conf. on Robotics and Automation, California, pp.1398-1404, 1991.

[9] Lidoris, G., Wollherr, D. & Buss, M., *Bayesian State Estimation and* Behavior *Selection for Autonomous Robot Exploration in Dynamic Environments*, IEEE International conf. on Intelligent robots and systems, France, 2008.

[10] Fulgenzi, C., Spalanzani, A. & Laugier, C., *Dynamic Obstacle Avoidance in Uncertain Environment Combining PVOs and Occupancy Grid*, IEEE International conf. on Robotics and Automation, Italy, 2007.

[11] Thrun, S., Probabilistic *Robotics*, The MIT Press, pp. 10-30, 2006.

[12] Li, Y. & He, K., *A Novel Obstacle Avoidance and Navigation Method of* Outdoor *Mobile Robot*, in proc. 12th International conf. on Advanced Robotics, pp.653-656, 2005.

[13] Foka, A. & Trahanias, E., *Predictive Control of Robot Velocity to Avoid Obstacles in Dynamic Environments*, IEEE International conf. on Intelligent robots and systems, Nevada, pp. 370-375, 2003.

[14] Budiharto, W., Purwanto, D. & Jazidie, A., *Indoor Navigation using ANFIS* controller *for Servant Robot*, in proc. IEEE 2nd International Conference on Computer Engineering and Its Application (ICCEA 2010), Bali-Indonesia, pp.582.586, 2010. DOI: 10.1109/ICCEA.2010.119.

[15] Budiharto, W., Purwanto, D. & Jazidie, A., *A Novel Method for Static and Moving Obstacle for Service robot using Bayesian Filtering*, IEEE 2nd International conf. on Advances in Computing, Control and Telecomunications Technology, 2-3 December, Jakarta, 2010, DOI: 10.1109/ACT.2010.51.

[16] Turk, M. & Pentland, A., *Face Recognition Using Eigenfaces*, In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[17] Yang, M., *Detecting Faces Images: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **24**(1), pp.34-58, 2002.

[18] Indian Face Database, http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase, last accessed 21 February 2010.

[19] ATT Face Database, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, last accessed 10 March 2010.

[20] OpenCV, http://sourceforge.net/projects/opencvlibrary, last accessed 10th November, 2011.