Multi-Sensor Based Collision Avoidance Algorithm for Mobile Robot

Abrar M. Alajlan, Marwah M. Almasri, Khaled M. Elleithy
Computer Science and Engineering Department
University of Bridgeport
Bridgeport, CT
aalajlan@my.bridgeport.edu, maalmasr@my.bridgeport.edu, elleithy@my.bridgeport.edu

Abstract—Collision Avoidance (CA) systems have been used in wide range of different robotics areas and had extraordinary success in minimizing the risk of collisions. It is a critical requirement in building mobile robot systems where they all featured some kind of obstacle detection techniques in order to avoid two or more objects from colliding. The purpose of this paper is to present an algorithm for performing collision avoidance in mobile robot that is relying on the use of low-cost ultrasonic with infrared sensors, and involving some other modules, so that it can be easily used in real-time robotic applications. The proposed algorithm is implemented in multiple scenarios with several obstacles placed in different locations around the robot. Our experimental run shows that the robot has been successfully detecting obstacles and avoiding collisions along its path.

Keywords- Collision avoidance algorithm, robotics control , obstacle detection, mobile robot

I. INTRODUCTION

Collision Avoidance (CA) systems have been used in wide range of different robotics areas and had extraordinary success in minimizing the risk of collisions. Collision avoidance systems are mostly applied in transportation systems such as aircraft traffic control, autonomous cars and underwater vehicles etc. Collision avoidance is a critical requirement in building mobile robot systems where they all featured some kind of obstacle detection techniques in order to avoid two or more objects from colliding (two mobile robots or one robot with an obstacle).

The main purpose of obstacle avoidance is to obtain a collision-free trajectory from the starting point to the target in monitoring environments. Obstacles can be divided into two types, static where the obstacle is predefined and has a fixed position and dynamic obstacle where the position is not pre-known and has uncertain motion patterns (moving objects). Detecting dynamic obstacles is more demanding than detecting static obstacles since the dynamic has a changeable direction and requires a prediction of dynamic obstacle position at every time step in order to achieve the requirement of a time-critical trajectory planning [1].

The dynamic obstacle algorithm is much more complex, since it does not only detect an object, but also consider some kind of measurements regarding the dimensions of the moving obstacle and distance between the mobile robot and object. As soon as these measurements have been determined, the obstacle detection algorithm needs to steer the robot around the object and continue toward the target [2]. However, this continuous procedures need to be performed while the robot is moving toward the original target [3].

In order to operate efficient collision avoidance technique, many successful mobile robot systems depend on the sensing capabilities and collision detection modules of the robot to obtain collision-free motion [4]. The collision position can be detected through some measurement coming from different modules such as cameras and GPS integrated with and variety of sensors like ultrasonic sensors and Infra-red sensors [5]. Based on fusion of these readings, the robot itself can make a decision to detect and prevent collisions while it is moving along the path, and thus mobility is not delayed until the detection process is updated [5].

In this brief, a fairly general algorithm is developed that has components of formation development, obstacles detection. Contributions of this paper is relying on the use of low-cost ultrasonic with infrared sensors, and involving some other modules, so that it can be easily used in real-time robotic applications.

The paper is organized as follows. Section 2 discusses the literature of collision avoidance techniques. Section 3 shows the hardware platform for the proposed approach. Section 4 presents the proposed algorithm of collision avoidance in robotic systems based on multiple sensing modules. The algorithm is then implemented and the overall system design and operation are discussed in section 5. The experimental results that demonstrate obstacle detection is presented in section 6. Lastly, section 7 offers conclusions.

II. RELATED WORK

Many collision avoidance algorithms have been proposed in the literature of robotics motion planning that allow the robot to reach its target without colliding with any obstacles that may exist in its path. Each algorithm differs in the way of avoiding static/dynamic obstacles. The concept of Artificial Potential Field (APF) is recently reported by M. Zohaib et al. in [6]. APF is a collision avoidance algorithm that always finds the shortest path from source to destination. It simply avoids an obstacle by generating a repulsive force from
obstacle to repel the robot and attractive force from target to attract the robot. Then, the total force on the robot is the sum of the attractive and the repulsive force. That is, the force is influenced by how far the robot is from obstacles and target. When the robot is close to target, its speed will be slow and vice versa [6]. Even though, APF has a very simple technique for collision avoidance, it is very sensitive to local minima in case of a symmetric environment [7].

The Vector Field Histogram (VFH) is a real-time obstacle avoidance algorithm that used to detect obstacles and avoid collisions while moving the robot toward its destination [2]. This method uses two-dimensional Cartesian histogram grid, where it divides the whole area around the robot into small sectors. In VFH, there is two-stage data reduction process in order to compute the desired commands. In the first stage, the two-dimensional histogram is converted to one-dimensional polar histogram. In the second stage, the algorithm selects the most suitable sector with low polar density and calculates the steering angle in that direction [8]. However, this algorithm is not suitable for detecting dynamic (moving) obstacles since its only studies the distance between the vehicle and obstacles while ignores obstacles velocities [9].

Sezer et al. proposed a new collision avoidance technique called Follow the Gap method (FGM). FGM avoids obstacles by finding the gap between them. Then, it calculates the gap angle and compares it with the threshold gap. Thus, the robot will follow the gap as long as it is greater than the threshold gap [10]. FGM performs three main computations. First, it calculates the gap array in order to find the maximum gap. Second, it calculates the center angle of the maximum gap to ensure safe path from obstacles center. Third, it calculates the final heading angle by combining both gap center angle with goal angle. Then, based on these calculations, the robot moves along final heading angle in order to avoid obstacles [11].

Several other obstacle avoidance algorithms are suitable for real-time application and will not be discussed here due to space limitations. Among the reported algorithms, the proposed one is intended to develop a collision avoidance mobile robot that is able to detect obstacles and avoid collisions. The mobile robot is equipped with multiple sensors and a microcontroller. These multiple sensors are used to get information about the surrounding area and then make a decision based on the sensors reading.

III. HARDWARE PLATFORM

.NET Micro Framework is an open source platform for embedded devices. Using Visual Studio and C#, developers can create embedded applications [12]. In addition, Microsoft has recently introduced the .NET Gadgeteer which is an open-source platform that enables using .NET Micro Framework and Visual Studio for combing the benefits of object-oriented programming and the assembly of small electronic devices [13]. .NET Gadgeteer is a standardized way to connect mainboards and modules. One of the most well-known companies that adapt the use of .NET Gadgeteer and .NET Micro Framework is GHI Electronics. GHI Electronics offers a variety of mainboards, sensor modules, and power modules [14].

FEZ Cerbot is a wheeled robot from GHI Electronics. It has a 168Mhz CPU, 1MB FLASH, 2 gears/motors, 16 configurable LEDs, 2 reflective sensors (infrared sensors), buzzer, and a four AA battery holder. The motors controls the speed and the direction up to 40V 3A. Also, FEZ Cerbot has six Gadgeteer sockets that can be used for serial modules, SPI modules, I2C modules, and I/O modules. FEZ Cerbot includes one USB Client cable for connecting to PC [14]. FEZ Cerbot robot is shown in Fig. 1.

The reflective sensor in FEZ Cerbot is an infrared sensor that is used for edge detection. It detects the presence of objects in a range. The sensor value changes when an object is detected within the range. The reflective sensor is shown in Fig. 2.

Distance US3 Module is used to measure the distance from an object. It is an ultrasonic sensor that measures the time between sending a sound wave and the sound echo back from that object and then calculates the distance. The range for this module is between 2cm and 400cm [14]. Distance US3 Module is presented in Fig. 3.
Furthermore, in order to capture images, a camera module is used. It is an USB Camera that streams images with a resolution of 320x240 [14]. The camera module is depicted in Fig. 4.

Fig. 4, Camera Module

GPS module is used to determine the current location coordinates and measures the distance. This module consists of a U-Blox Neo-6M GPS module and an antenna [14]. Fig. 5 shows the GPS module.

Fig. 5, GPS Module.

IV. THE PROPOSED APPROACH

This section describes the proposed approach for collision avoidance based on different types of sensor modules. Based on the output of these modules, the robot will avoid obstacles by detecting them and change its direction accordingly. The sensor modules used are infrared sensor for edge section, distance module which is an ultrasonic sensor that is used to measure the distance between the robot and the detected object, camera module that is used to detect objects and measure the object dimensions and the distance, and the GPS modules to determine the coordinates and the distance between two points. Four algorithms are proposed in this paper. Three algorithms handle the sensor readings where the fourth algorithm is the main algorithm that incorporates the other algorithms.

1. **Edge detection and distance measurement**

In order to detect the edges, the reflective sensors (infrared sensors) on the robot are used. Two infrared sensors: one on the left and one on the right. After getting these sensors’ readings, the readings will be compared with a threshold value within the range of detection. If both infrared sensors values are less than the threshold, then an obstacle is detected. Otherwise, use the Distance US3 Module to measure the distance from an object within 2cm to 40 cm range. GetDistanceInCentimeters() function is used to convert the sensor readings into centimeters. The detailed algorithm is shown in Algorithm 1.

2. **Camera-based Object detection Algorithm**

Camera module is used to capture images. These images are processed to extract and detect the objects from the image. After detecting objects, the image will be converted to grayscale image. After that, thresholding is performed in order to isolate the object from the background followed by noise elimination. After extracting the object from the image and isolate it from the back ground, the object’s dimension will be measured. By knowing the coordinates of the object in 2-D environment, then the distance between the robot and the object is obtained [15]. Algorithm 2 summaries the Camera-based Object detection algorithm.

<table>
<thead>
<tr>
<th>Algorithm 1: Edge detection and distance measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> L: Left infrared sensor, R: Right infrared sensor, a: ultrasonic sensor reading</td>
</tr>
<tr>
<td><strong>Output:</strong> β: Edge detection, gd: distance in cm</td>
</tr>
</tbody>
</table>

```
begin
set L
set R
// threshold=10
if ((L< threshold) &&(R< threshold))
then set β== True
else
   if (a== set value)
      // Convert reading to cm
      then set gd= GetDistanceInCentimeters()
      return gd
end if
end if
end else
end
```

Algorithm 1: Edge detection and distance measurement
Algorithm 2: Camera-based Object detection

Input: I: Image from Camera, gr: convert RGB to Grayscale, thr: thresholding, noise: Eliminate all noises, di: Measure object dimensions
Output: \( \Delta d\): distance between object and camera

begin
  set I // using Image processing
  Extract obj
  // convert color image to grayscale image
  Set gr
  // for separating the object from the background
  Set thr
  // Eliminate all noises from the image
  Set noise
  // Measure the dimensions of obj
  Set di
  // (x1, x2) coordinates of obj1, (y1, y2) coordinates of obj2
  Set \( \Delta d = \sqrt{((\Delta x)^2 + (\Delta y)^2)} = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}) \)
  return \( \Delta d \)
end

Algorithm 3: GPS-based distance measurement

Input: pos1: position1, pos2: position2
Output: \( d\): Distance between two positions

begin
  set pos1 and pos2 // convert degree to radians
  Dlat1=pos1.lat *PI/180
  Dlong1=pos1.long *PI/180
  Dlat2=pos2.lat *PI/180
  Dlong2=pos2.long *PI/180
  \( \Delta \text{lat} = \text{dlat2-dlat1} \)
  \( \Delta \text{long} = \text{dlong2-dlong1} \)
  // convert from radians to cm using Haversine formula
  \( a = \sin^2(\Delta \text{lat}) + \cos(\text{dlat1}) \cdot \cos(\text{dlat2}) \cdot \sin^2(\Delta \text{long}/2) \)
  \( c = 2 \cdot \tan(\sqrt{a}/\sqrt{1-a}) \)
  \( d = R \cdot c \) // R is the earth radius
  return \( d \)
end

Algorithm 4: Collision avoidance algorithm

Input: s: Motor speed, gd: distance in cm by ultrasonic sensor, \( \Delta d\): distance between object and camera, d: Distance between two positions by GPS
Output: Collision avoidance

begin
  set s
  While (true)
  do Edge detection and distance measurement algorithm
    if (\( \beta\) != True)
      do Camera-based Object detection
      do GPS-based distance measurement
      Dist=Fusion (gd, \( \Delta d\), d) // Collision avoidance Decision
      if (Dist >= 5 && Dist <= 30)
        set motorSpeed(0,0) // stop
        set Delay =1000 ms
        set motorSpeed(-s,-s) // Backup
        set Delay =1000 ms
        set motorSpeed(s,-s) // Turn
      end else
      set motorSpeed(s,s) // Move Forward
    end else
  end while
end

Algorithm 3: GPS-based distance measurement

3. GPS-based distance measurement Algorithm

By using GPS module, the robot location is determined. After taking the readings of the robot and the object, a formula is used to convert degree to radians and later to centimeters using Haversine formula. Knowing the earth radius, the distance between the robot and the object is calculated. Algorithm 3, explains the process in more details.

4. Collision avoidance algorithm

This algorithm is the main algorithm as it incorporates the other algorithms’ output as an input. The distance between the robot and the object is measured by the Distance module. The reflective sensor is used for edge detection. The camera module is used to detect objects from the image and then measure the object’s dimensions and the distance between the robot and the object as well. GPS module is used to measure the distance between the robot and the object. After obtaining all these data, sensor data fusion is done and the collision avoidance decision is made. If the robot detects an object in the sensing range, it will stop, wait for a second, back up, wait for another second and then turn. Finally the sensor moves forward. Algorithm 4 describes the process in details. The flowchart of the algorithm is shown in Fig. 6.

V. MOBILE ROBOT EXPERIMENTAL SETUP

The collision avoidance algorithm was tested on FEZ Cerbot robot from GHI Electronics Company. This robot has
been equipped with multiple sensors as follow. Two infrared sensors at the front (left and right), ultrasonic sensor (Distance US3 module), camera module, and GPS module as presented in Fig. 7. The algorithm were implemented on a Windows platform using .NET. The parameters used in this experiment are defined in Table I.

Table. I: parameters used in the experiment

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor speed</td>
<td>70</td>
</tr>
<tr>
<td>Sensor Range</td>
<td>2 cm – 40 cm</td>
</tr>
<tr>
<td>Threshold for edge detection</td>
<td>10</td>
</tr>
</tbody>
</table>

VI. EXPERIMENTAL RESULTS

The main goal of this work is to detect objects within the range of detection. We deployed multiple sensors in order to enhance the overall performance of the robot by improving its reliability and robustness. The most accurate measurement among them is obtained by performing sensor data fusion. After data fusion is done, the robot will be avoiding the obstacles and turning away from any detected obstacles. The snapshots of an experimental run are shown in Fig. 8.

As shown in Fig. 8, the robot starts sensing the environment by different types of sensors and moves forward due to no presence of objects in its way. Once the object is detected, the distance is computed between the robot and the object based on these sensor modules. Depending on the calculations, the robot will stop and then back up. After that, it will turn to avoid the object on its path and finally move forward. As depicted in Fig. 8 (3&4), the robot detects another object on its way and again it avoids it and then turns. This experiment was implemented in multiple scenarios with several obstacles placed in different locations around the robot. Our experimental results show that the robot has been successfully detecting obstacles and avoiding collisions along its path.
VII. CONCLUSIONS

An obstacle avoidance methodology using multiple sensors for the mobile robot was presented and has been developed and tested on our experimental mobile robot FEZ Cerbot. The proposed framework is a real time and portable for any environment with various shapes of obstacles it encounters during navigation. The framework was successfully tested with various shapes of obstacles. The distance fusion calculated using the collision avoidance algorithm shows the improved performance in detecting objects and avoiding collision.

REFERENCES


