

Article

# An Optimal and Energy Efficient Multi-Sensor Collision-Free Path Planning Algorithm for a Mobile Robot in Dynamic Environments

Abrar Alajlan, Khaled Elleithy \*, Marwah Almasri and Tarek Sobh

Department of Computer Science, University of Bridgeport, 126 Park Avenue, Bridgeport, CT 06604, USA; aalajlan@my.bridgeport.edu (A.A.); maalmasr@my.bridgeport.edu (M.A.); sobh@bridgeport.edu (T.S.)

\* Correspondence: elleithy@bridgeport.edu; Tel.: +1-(203)-576-4703

Academic Editor: Huosheng Hu

Received: 4 December 2016; Accepted: 28 March 2017; Published: 31 March 2017

**Abstract:** There has been a remarkable growth in many different real-time systems in the area of autonomous mobile robots. This paper focuses on the collaboration of efficient multi-sensor systems to create new optimal motion planning for mobile robots. A proposed algorithm is used based on a new model to produce the shortest and most energy-efficient path from a given initial point to a goal point. The distance and time traveled, in addition to the consumed energy, have an asymptotic complexity of  $O(n \log n)$ , where  $n$  is the number of obstacles. Real time experiments are performed to demonstrate the accuracy and energy efficiency of the proposed motion planning algorithm.

**Keywords:** motion planning; collision avoidance; safe navigation; autonomous control; mobile robot

---

## 1. Introduction

The robotic motion planning research domain has received a considerable amount of attention over the last decade, as robots are becoming a critical part of the automobile industry, underwater vehicles, and airport terminals. Collision avoidance is one of the fundamental requirements in designing mobile robot systems, where almost all of the mobile robot applications feature some kind of obstacle detection methods.

Path planning in mobile robots is categorized as global or local, based on the tested environment. In global path planning, the surroundings are predefined and the path is known in advance. Conversely, in local path planning, the environment is unknown, and different types of modules are used to detect obstacles and avoid collisions [1].

In the real world, the autonomous vehicles have to safely operate in unpredictable environments and reach their destination, regardless of unanticipated changes in the tested area. One of the most significant characteristics of using autonomous vehicles is to attain some information about the surroundings through different types of sensors taking measurements, and then interpret the gathered information to produce readable data for the control system. Several studies have been introduced in the literature to extract the environmental details from the gathered data [2].

In a collision-free path mode, the resulting motions depend upon the sensing capabilities and the actual position of the mobile robot [3]. In addition, an obstacle's exact location can be determined by the measurements obtained from the readings of multiple sensors [4]. Since a single sensor is not capable of completing the task, employing multiple sensors is important. Information from different sensors is obtained and combined to find the location of the robot, detect obstacles, and avoid collision. In order to perform these tasks, the robot has to have an outstanding sensory system, a strong mechanical formation, and a sturdy controlling system.

The work presented in this paper demonstrates the use of multiple sensors in the mobile robot to generate optimal motion planning, where the robot attains the surrounding information through the sensors' readings.

Novel optimal motion planning is introduced to optimize the energy consumption during motions where an algorithm has been used to create a collision-free path. The path planner saves the energy by computing the shortest path (least-cost path) from a given position to the target position [5].

This paper is structured as follows. Section 2 provides a literature overview of collision avoidance techniques. Section 3 demonstrates the multi-sensory system used in the proposed model. A detailed description of the proposed approach is presented in Section 4. The real-time experiments are presented in Section 5. The experimental results and data analysis are discussed in Section 6. Finally, Section 7 offers conclusions.

## 2. Related Work

Path planning in mobile robots can be classified into two types; global path planning and local path planning. In global path planning, an obstacle shape is known and the mobile robot has a predefined motion. On the other hand, local path planning provides a collision-free path by using geometric information such as an obstacle's velocity and the distance between the robot and any obstacles along its path [6]. Several algorithms have been proposed in the literature of mobile robotics studies for producing a collision-free path for mobile robot navigation. These include the follow the gap method, artificial potential field, vector field histogram, Bug algorithm [7], Vision sensor-based method (VSB) [8], and New Hybrid Navigation Algorithm (NHNA) [9], etc.

The Follow the Gap Method (FGM) is a well-known collision avoidance algorithm that aims to identify the gap angle between the obstacles. In order to compare it with a specified threshold [10]. FGM assumes that the robot and obstacles are circular objects, then measures the robot's dimensions in order to add it into an obstacle's dimension for calculating the path. This process can be done as follows: it computes the gap array that stores all of the distances between obstacles and the robot in order to discover the maximum gap between obstacles, calculates the center angle of the maximum gap to guarantee a safe path from the obstacle's center, and then calculates the final heading angle by combining the gap center angle with the goal angle. After performing the entire process above, the robot avoids the obstacles by moving towards the final heading angle.

Another collision avoidance algorithm based on the concept of the potential field has been proposed in [11], by M. Zohaib et al. It is called the Artificial Potential Field (APF). In this method, the obstacles generate a repulsive force to repel the robot, while the target produces an attractive force to attract the robot. The total force of the robot is interpreted by summing up the repulsive forces from the obstacles and the attractive force from the target. The total force is affected by how far the robot is from the obstacles and its destination.

The Vector Field Histogram (VFH) is a real-world obstacle avoidance method that was designed to reduce the limitations of the Artificial Potential Field (APF) method [12]. It is utilized to detect obstacles and avoid a collision as the robot travels toward the destination [13]. VFH has a two-dimensional Cartesian histogram grid, and the world model is divided into small sectors. VFH employs a two-stage of data reduction process in order to compute the desired commands. A two-dimensional histogram is converted into a one-dimensional histogram, and then into a one-dimensional polar histogram.

The algorithm selects the most appropriate sector that has a low polar density, and then calculates the steering angle for that direction [14].

Among the reported algorithms, very few methodologies have been devoted to using a low computational power with reasonable accuracy. Table 1 highlights some important findings, with a focus on the limitation of each method.

### 3. Multisensory System

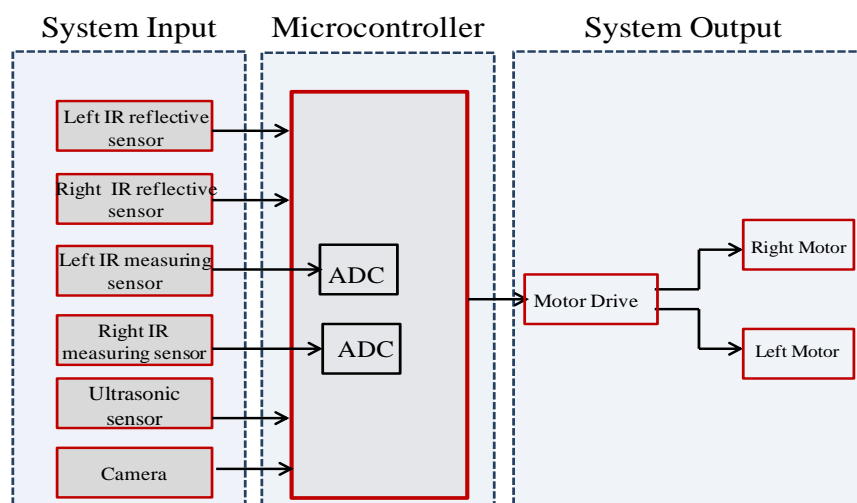
In some applications, the operating environments can be really adverse, depending on the sensors used. Since all sensors feature limitations, the integration of supplied information from different types of sensors and utilizing the advantages of each type is the most reliable solution. In this scheme, different types of sensors are used to create a complementary multi-sensors system that has the ability to produce optimal path planning while avoiding collisions, thus delivering reliable measurements.

#### 3.1. Functional Block Diagram

The block diagram of the proposed motion planning approach is provided in Figure 1. It utilizes multiple sensors: two infrared reflective sensors for edge detection, two infrared measuring sensors, an ultrasonic sensor, and a camera for obstacle detection. The output of the IR measuring distance sensors is analog in nature, as it is based on the amount of reflected light. Thus, the analog output needs to be processed to generate digital signals containing information about the obstacle’s distance using an Analog to Digital Converter (ADC). Then, the microcontroller processes all of the data acquired from the mounted sensors and sends control signals to the motor drive in order to perform the required moves. Modeling of the robot’s environment and the typical characteristics of the various sensors used in this work are described in further detail in the next section.

**Table 1.** Comparison of path planning methods.

	Detection of Critical Shaped Obstacles	Computational Cost	Path Planning	Obstacle Avoidance
<b>Follow the Gap Method</b>	Fails	High cost for calculation	Always selects shortest path	Able to avoid symmetric obstacles
<b>Artificial Potential Field</b>	Fails	High cost for modeling	Always selects the shortest path	Fails to avoid symmetric obstacles
<b>Vector Field Histogram (VFH)</b>	Fails	High cost for modeling and histogram generation	Always selects the shortest path	Fails to avoid symmetric obstacles
<b>Bug Algorithm</b>	Fails	High cost for path planning		May take the robot away from destination
<b>Vision sensor based method</b>	Fails	High cost for vision		
<b>New Hybrid Navigation Algorithm</b>	Fails	High cost for map and path generation		



**Figure 1.** Functional Block Diagram of the proposed model.

### 3.1.1. Infrared Reflective Sensors

Infrared reflective sensors work best for detecting lines and edges, as they measure the light reflection from the transmitted IR beam. Each sensor is composed of two parts: the emitter (transmitter), used to transmit the light waves through the surface; and the detector, used to sense the reflected light and measure the output voltage based on the amount of reflected light.

### 3.1.2. Infrared Distance Measuring Sensors

The main purpose of using the infrared distance measuring sensor is to detect the reflected light when hitting an object. The amount of received light is then computed using a triangulation method to calculate the distance to the object. The output of this sensor is an analog voltage that has to be converted to a digital form at the microcontroller, in order to identify the distance to the object.

### 3.1.3. Ultrasonic Sensor

The ultrasonic sensor is used for obstacle detection. It transmits the sound waves, which makes it ideal for detecting an object in poor lighting areas as the performance is independent of the object's color or surface properties. The ultrasonic sensor detects obstacles by transmitting high frequency sound waves through the surface, and in the case of the existing object, the waves collide with the object and bounce back to sensor. Then, the time interval between sending the signal and receiving the echo is calculated by the sensor, to determine how far the object is from the sensor. The ultrasonic sensor shows its true strength for applications that require precise measurements between stationary and moving objects.

### 3.1.4. Camera

Video cameras are one of the most commonly used sensors in a wide range of applications, such as the detection, tracking, and recognition of objects. A single onboard camera has been mounted at the front of the robot to detect and track objects. The object detection process can be divided into four main procedures, as illustrated in Figure 2 and the following four algorithmic steps:

1. Eliminate the ego-motion effect of the camera and extract a Region of Interest (ROI). Two images of the same scene are linked by a nonsingular linear transformation (previous image and current image), so the effect of ego motion can be eliminated.
2. Detect the foreground object and perform video segmentation. The first frame of the video is used as the background, where the intensity and color information for the background subtractions are considered to improve the accuracy of foreground segmentation.
3. Calculate the foreground objects and connect the foreground points as a blob using a blob analysis block. The blob analysis block is used to calculate statistics for labeled regions in a binary image, where a blob library for OpenCV is used to detect the connected foreground region.
4. Display the detected object in the image space.

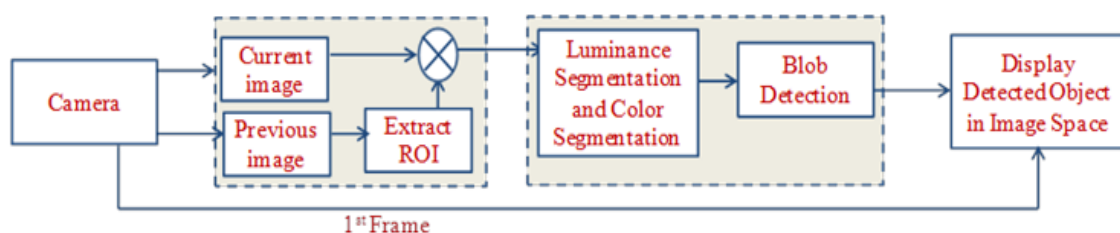


Figure 2. Block diagram of object detection by camera.

### 3.1.5. Microcontroller

The microcontroller works as the robot's brain, where it is responsible for creating control signals and making decisions. The microcontroller has to process all of the data acquired from sensors and performs all of the calculations to make the right decisions. The analog to digital converter is used to convert the analog voltage output produced by the infrared distance measuring sensors to a microcontroller-readable form (digital number).

### 3.1.6. Motor Drive

The motor drive receives the control signals from the microcontroller and performs an action accordingly. It consists of two motors: the left motor and right motor, to control the motion and adjust the speed.

The integration of the information supplied by multiple sensors can be the best solution to overcome the spatial uncertainty of unknown environments in several advanced robotics navigation applications.

## 4. Description of Proposed Approach

Figure 3 presents a flowchart describing the proposed motion planning approach. The proposed model deploys a searching algorithm to find the shortest path from a given point to the target. It has been widely used in path finding for its reduced search space. It uses a best first search, where it takes an input, evaluates a number of possible paths, and produces the least-cost path from a given source to the destination [15]. As the robot follows the least-cost path, it maintains a sorted queue of alternate paths along the way, in order to abandon a higher cost path segment at any point in time and use the lower-cost segment instead. This procedure continues until the target is reached [15]. The cost of the path can be calculated based on the distance traveled from a given point to another, as follows:

$$g(x) = g(x - 1) + Dx \quad (1)$$

where  $g(x - 1)$  is the cost of the previous path segment and  $Dx$  is the traveling distance between  $x - 1$  and  $x$ . Then, the least-cost path from a start point to the target can be determined based on the cost of each path segment, as follows:

$$f(x) = g(x) + h(x) \quad (2)$$

where  $h(x)$  is a heuristic estimated cost of the path from the current point to the goal point. The algorithm used in this work is characterized by a high searching efficiency and completeness.

When the shortest path from the initial position to the target has been determined, the proposed collision avoidance model performs two processes: edge detection and obstacle detection. At the initial position, the robot starts sensing the environments for edges through left and right reflective sensors, to avoid moving over the edge. In case of possible collision, the reflective values are compared with a predefined threshold and the robot will perform an action accordingly.

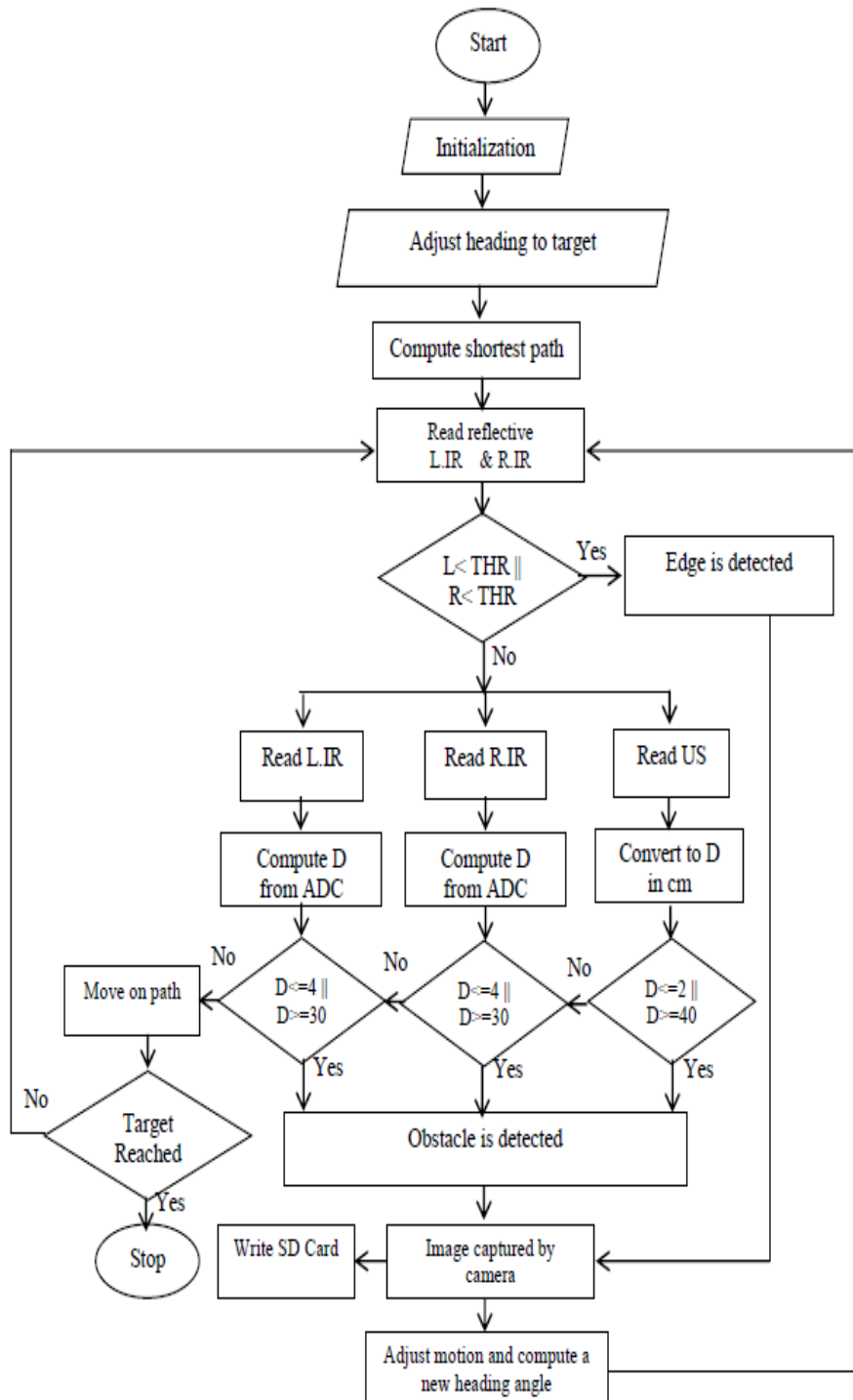
On the other hand, if there is no detected edge, the difference between the left and right sensor readings is close to zero. Therefore, both motors move at the same speed (moving forward). Then, the robot performs obstacle detection in order to avoid any imminent collisions. Two infrared distance measuring sensors, an ultrasonic sensor, and a camera are used for obstacle detection. The microcontroller converts all of the sensors readings into a measurable form (distances in centimeters), to compare them with their detection ranges. If an object is detected through any sensor, the robot spins around the object and moves forward. Thus, the microcontroller decision is based on the readings of the sensors.

## 5. Real Time Experiments

In this work, a FEZ Cerbot robot (GHI Electronics, Madison Heights, MI, USA) is used to test the proposed technique. Figure 4 shows the entire structured mobile robot, including the infrared reflective sensors, infrared distance measuring sensors, ultrasonic sensor, camera, and robotic platform.

The infrared and ultrasonic sensors are used in combination, where the benefits of one compensate for the limitations of the other. For example, in unknown environments, the ultrasonic sensor is utilized to verify the surface properties in order to understand the distance measurements from the infrared sensor.

A set of experiments were executed in multiple scenarios, with different shapes of obstacles placed in multiple locations around the robot. The experiments were carried out as follows.



(a) Flowchart of the proposed motion planning algorithm.



**Algorithm 1:** Dynamic Collision Avoidance Algorithm

```

Input: L.ref: Left reflective sensor; R.ref: Right reflective sensor; US: Ultrasonic sensor; L.IR: Left measuring distance sensor; R.IR: Right measuring distance sensor; S: Motor speed; thr: Threshold; t:time;
Output:  $\alpha$ : Edge is detected ;  $\beta$ : Obstacle is detected
BEGIN
  INIT s //initialize the speed
  GET source.position == start //get start point
  GET goal.position == goal //get goal point
  WHILE (true)
    IF start == goal // compare if the start point equal to goal point
      THEN Goal reached
    ELSE WHILE (start != goal) DO
      ADJUST heading to goal
      COMPUTE shortest path to goal
      READ L.ref // Read left reflective sensor
      READ R.ref // Read left reflective sensor
      IF ( L.ref < thr && R.ref < thr ) //Compare sensor reading and right sensor reading with threshold
        THEN SET  $\beta$  ==true // Edge is detected if both readings are less than the threshold.
      ELSE
        READ US // read ultrasonic sensor
        CALL GetDistanceInCentimeters (US) // convert the reading to centimeter
        READ L.IR // read left infrared sensor
        CALL AnalogToDistance (L.IR) // convert analog reading to digital reading
        READ R.IR // read right infrared sensor
        CALL AnalogToDistance (R.IR) // convert analog reading to digital reading
        IF ( (US >= 5 && US <= 30) || (L.IR >= 4 && L.IR <= 30) || (R.IR >= 4 && R.IR <= 30) )
          THEN CALL SetMotorSpeed (0,0) // the left motor speed and right motor speed to stop
          SET t = 1000ms // set timer for 1 sec
          CALL SetMotorSpeed (-s,s) // turn left
          READ L.IR // read left infrared sensor
          CALL AnalogToDistance (L.IR) // convert analog reading to digital reading
          CALL SetMotorSpeed (s,-s) // turn right
          READ R.IR // read right infrared sensor
          CALL AnalogToDistance (R.IR) // convert analog reading to digital reading
          IF (L.IR >= R.IR) // if the left distance is smaller than right distance
            THEN CALL SetMotorSpeed (s,s) // move forward
          ELSE
            CALL SetMotorSpeed (-s,s) // turn left
            CALL SetMotorSpeed (s,s) // move forward
        ELSE
          CALL SetMotorSpeed (s,s) // move forward .. no object detected
        ENDIF
      ENDIF
    ENDWHILE
  ENDIF
ENDWHILE

```

(b) Pseudo code of the motion planning algorithm.

Figure 3. Motion Planning Algorithm.

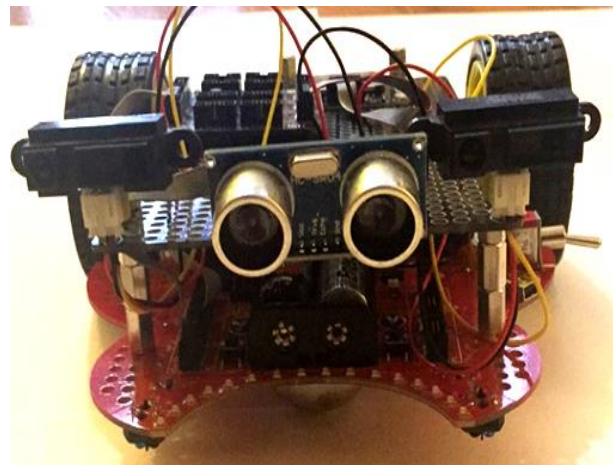


Figure 4. The test-bed prototype of the FEZ Cerbot.

### 5.1. Avoiding Static and Dynamic Obstacles

The robot is equipped with multiple sensors to create an alert if an object is detected while moving along its path. For each experiment, the robot starts sensing the environment for objects, and when an object is detected, the robot adjusts the motion and computes the new heading position. When an object is detected through the right infrared sensor, the maximum amount of light is reflected back to the right infrared sensor, while less light is reflected to the left infrared sensor. The difference between the right and left readings is positive, so the right motor speeds up and the left motor slows down, and then the robot turns to the left. On the other hand, when the obstacle is detected through the left infrared sensor, the difference between the right and left readings is negative. Thus, the right motor slows down and the left motor speeds up, and then the robot turns to the right.

Consider an environment which has static obstacles of different shapes. The environment contains polygonal, convex, curved, and transparent obstacles of different shapes and sizes. The first experiment was conducted to examine the ability of the infrared and ultrasonic sensors to detect static obstacles and avoid collisions, as illustrated in Figure 5.



**Figure 5.** Obstacle detection in a static environment.

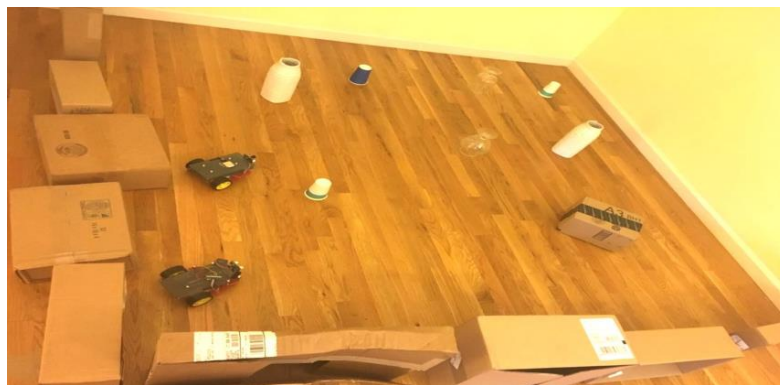
The second set of experiments were conducted to examine the ability to detect critical-shaped obstacles. The existing algorithms fail in avoiding U-shaped obstacles, as APF and FGM are unable to make a decision when the distance to the left and right sidewalls is the same, while it is considered a dead-end scenario for VFH. Figure 6 shows the trajectory of the robot when detecting a U-shaped obstacle. The robot senses the side walls, calculates the distance to the front wall, and steers it inside as it is greater than a predefined threshold. As the front wall is detected, the robot turns about  $90^\circ$  to avoid it, and it then turns about  $90^\circ$  again when the side wall is detected. After avoiding a U-shaped obstacle, the robot continues its motion in that direction.

Another set of experiments were performed in a partially dynamic environment, where it was comprised of multiple static objects and another robot served as a dynamic object. Figure 7 shows a snapshot of the partially dynamic environment. The experiments were performed in an open area with two robots, where each robot was equipped with a proximity sensor to detect the other and to measure their relative distance.





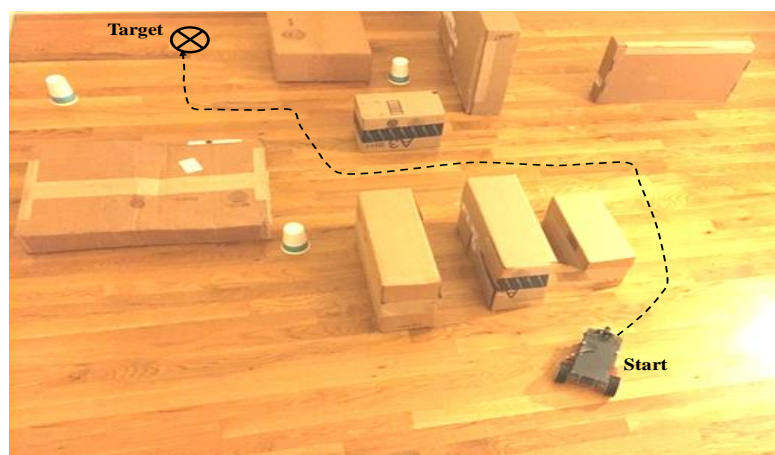
**Figure 6.** The trajectory of the robot when detecting a U-shaped obstacle.



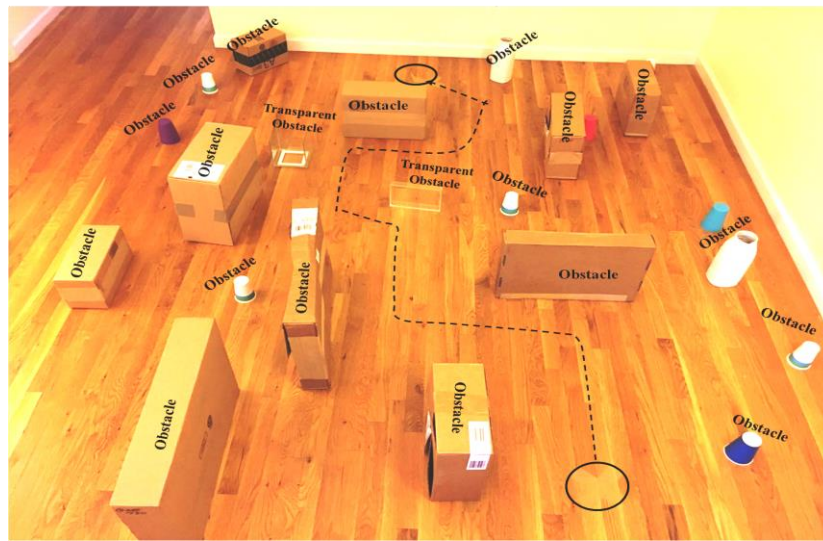
**Figure 7.** Obstacle detection in a partially dynamic environment.

### 5.2. Collision-Free Path in Simple and Complex Environment

The experiments were carried out to find an optimal collision path in simple and complex dynamic environments, where the robot started from a given initial position and needed to travel toward a predefined goal position using the shortest path. The two experiments are illustrated in Figures 8 and 9, respectively.



**Figure 8.** Motion planning based on the proposed algorithm in a simple environment.



**Figure 9.** Motion planning based on the proposed algorithm in a complex environment.

The experiments were performed to demonstrate the effectiveness of the proposed algorithm in terms of following the optimal path from the source to the destination. The travel distance, travel time, energy cost, and the robot velocity in simple and complex optimal collision-free paths are presented in Table 2.

**Table 2.** Experimental results of a collision-free path in simple and complex environments.

Environmental Type	Number of Detected Object	Traveled Distance (m)	Traveled Time (s)	Energy Cost (J)	Robot Velocity (ms)
Simple	4	1.27	164	629	0.7
Complex	7	1.76	189	940	0.7

The experimental findings demonstrate that the proposed method provides a safer and smoother navigation in the presence of obstacles.

## 6. Result and Discussion

The main contribution of this work is to produce an accurate, energy efficient, and fast planning model for a wheeled mobile robot. The performance evaluation criteria include optimizing the traveled distance, the traveled time, and the energy consumption, while avoiding all static and dynamic obstacles. The results of using multiple sensors are discussed in the following subsections.

Data acquisition (DAQ) was used to monitor the energy consumption while the robot is traveling from a given initial position to its goal, avoiding different types of obstacles. To illustrate the results, further experiments were conducted with a different number of obstacles, as shown in Figure 10. The energy consumption increases as the number of obstacles increases. The mobility energy is 397 (J) to cover 0.71 (m), which is the minimum distance from the source to the destination when no obstacles were found in the path. The energy consumption has an asymptotic complexity of  $O(n \log n)$ , where  $n$  is the number of obstacles.

The time-distance graph of the proposed algorithm is depicted in Figure 11. The minimum distance from a given initial position to the goal is 0.71 m, which is covered in 149 s when no obstacle is detected along the path. Considering obstacles in the path, the time and travel distance will increase as the robot consumes more time and distance to spin around the obstacles and avoid collisions. The distance and time traveled have an asymptotic complexity of  $O(n \log n)$ , where  $n$  is the number of obstacles.

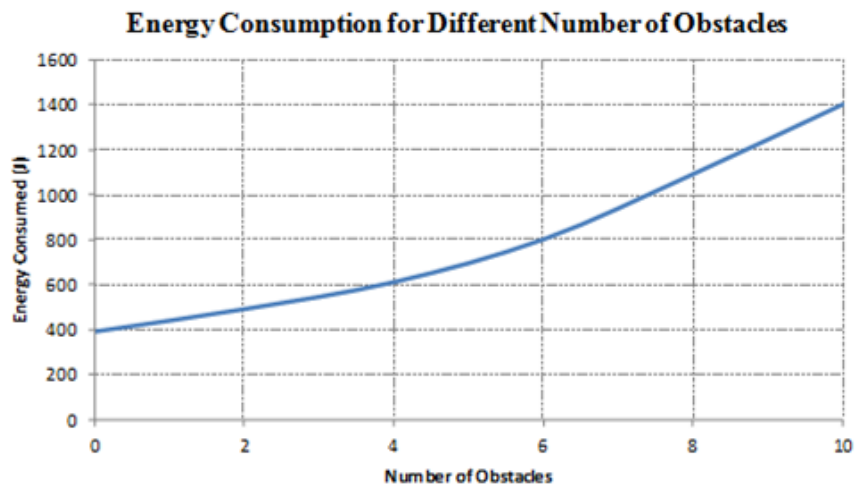


Figure 10. Energy Consumption of a Different Number of Obstacles.

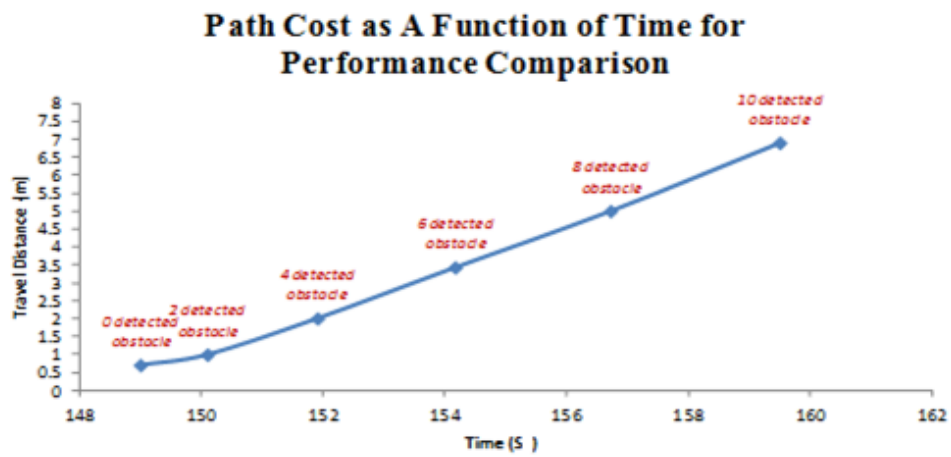


Figure 11. Path cost as a function of time for performance comparison.

### 6.1. Infrared Distance Measuring Sensors

The infrared sensor readings are illustrated in Figure 12.

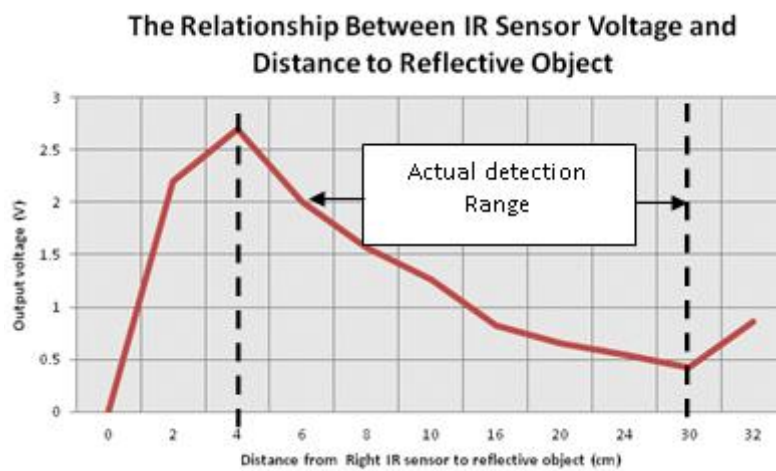


Figure 12. The right infrared sensor reading during the experiment and the distance to the reflective object.

The output voltage of the sensor depends upon the distance measured. As the distance increases, the output voltage decreases (inversely proportional). The infrared sensors produce an analog voltage relative to the distance. Thus, an analog to digital converter (ADC) is needed to find the distance to the reflective object.

The ADC of our robot is a 10-bit ADC, implying that it has the ability to detect  $2^{10}$  analog levels and the value returned by ADC is a ratio-metric value, which is a ratio between the output voltage and analog levels as in:

$$ADC_{val} = \frac{ADC_{level} \times Out_{vol}}{Opt_{vol}} \tag{3}$$

where  $ADC_{level}$  is the analog detecting level (=1023 in our robot),  $Out_{vol}$  refers to the analog output voltage at any given distance, and  $opt_{vol}$  refers to the system operating voltage (5 V in our robot). In order to find the distance from the analog value, the corrected inverse of the analog voltage related to ADC is computed in Figure 13.

$$D = \left( \frac{1}{cs_0 \times ADC_{value} + cs_1} \right) - K \tag{4}$$

where  $cs_0$  and  $cs_1$  are positive constant coefficient parameters,  $ADC_{value}$  is the sensor's reading at any point in time, and  $K$  is the corrective constant. Table 3 shows the readings of the sensors at different times during the experiment.

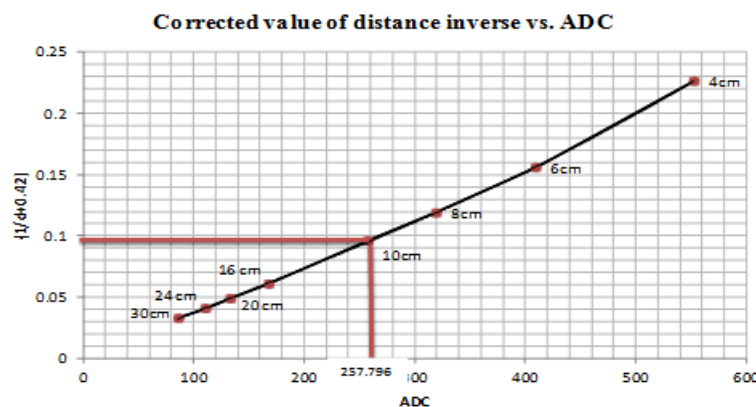


Figure 13. The corrected inverse of distance and  $ADC_{val}$ .

Table 3. Infrared distance measuring sensors at different experimental times.

Time	Right Infra-Red (RIR)	Left Infra-Red (LIR)	Comments
$t_0$	38.874	40.92	no object is detected
$t_5$	151.404	157.542	object is detected through RIR and LIR
$t_{10}$	20.46	28.644	no object is detected
$t_{18}$	40.92	42.966	no object is detected
$t_{22}$	216.876	40.92	object is detected through RIR
$t_{31}$	24.552	32.9406	no object is detected
$t_{39}$	36.828	43.3752	no object is detected
$t_{41}$	167.772	40.92	object is detected through RIR
$t_{56}$	26.598	20.6646	no object is detected
$t_{64}$	30.69	26.598	no object is detected
$t_{72}$	24.552	409.2	object is detected through LIR
$t_{81}$	36.828	30.69	no object is detected
$t_{93}$	40.92	257.796	object is detected through LIR
$t_{102}$	20.46	22.506	no object is detected
$t_{118}$	32.736	40.92	no object is detected
$t_{126}$	26.598	42.966	no object is detected

The Sharp GP2D120 (Sharp Microelectronics, Osaka, Japan) has a detection range between 4 and 30 cm, and the corresponding ADC values range between 530 (for a distance of 4 cm) and 80 (for a distance of 30 cm). Both sensors return an incorrect value for a distance out of their detection range (less than 4 cm and greater than 30 cm). For example, at time t0, no object is detected, as both sensors' values are out of the detection range. At time t5, both sensors' values are greater than 80 and less than 530, which indicates that a front object is detected by both sensors. At time t22, the right infrared sensor detects an object, while the left infrared does not, which indicates that the object is positioned at the front right of the robot.

6.2. Ultrasonic Sensor

The ultrasonic sensor is used to measure the distance to the object. In order to improve the angle uncertainty in the sonar data, it is suggested that one assumes that there is no information available about the obstacles. A Bayesian approach can be employed with the assumption of a prior distribution for the presence of unknown obstacles [16]. An iteration process is used to estimate the probability density function to describe the random obstacle events that might occur during the movement of the robot. According to Figure 14, Distance US3 modules produce a fairly accurate representation of the object location when compared with the real distance. However, the values for distances of less than 2 cm are not reliable, since they are situated outside of the sensor's detection range (2 cm to 40 cm). Two tests are performed for the Distance US3 module and the results are illustrated in Figure 15.

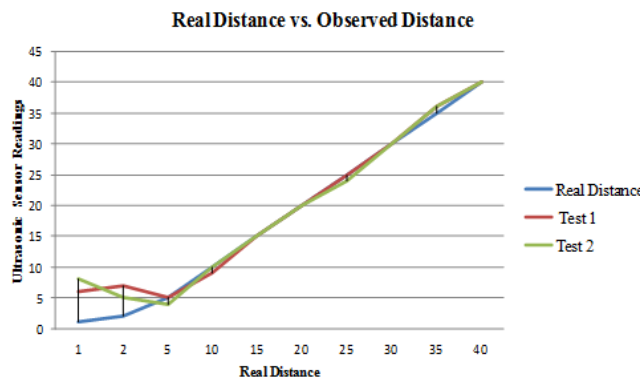


Figure 14. Testing the ultrasonic sensor for measuring the distance to the object.

Figure 15 shows the distance to the object at different experimental times. The distance to the object has been computed based on all of the sensors' readings, to determine how far the object is from the robot.

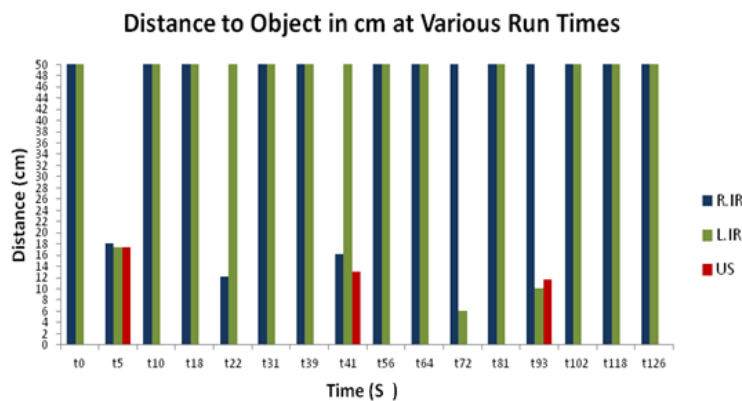


Figure 15. Distance to the object at different experiment times.



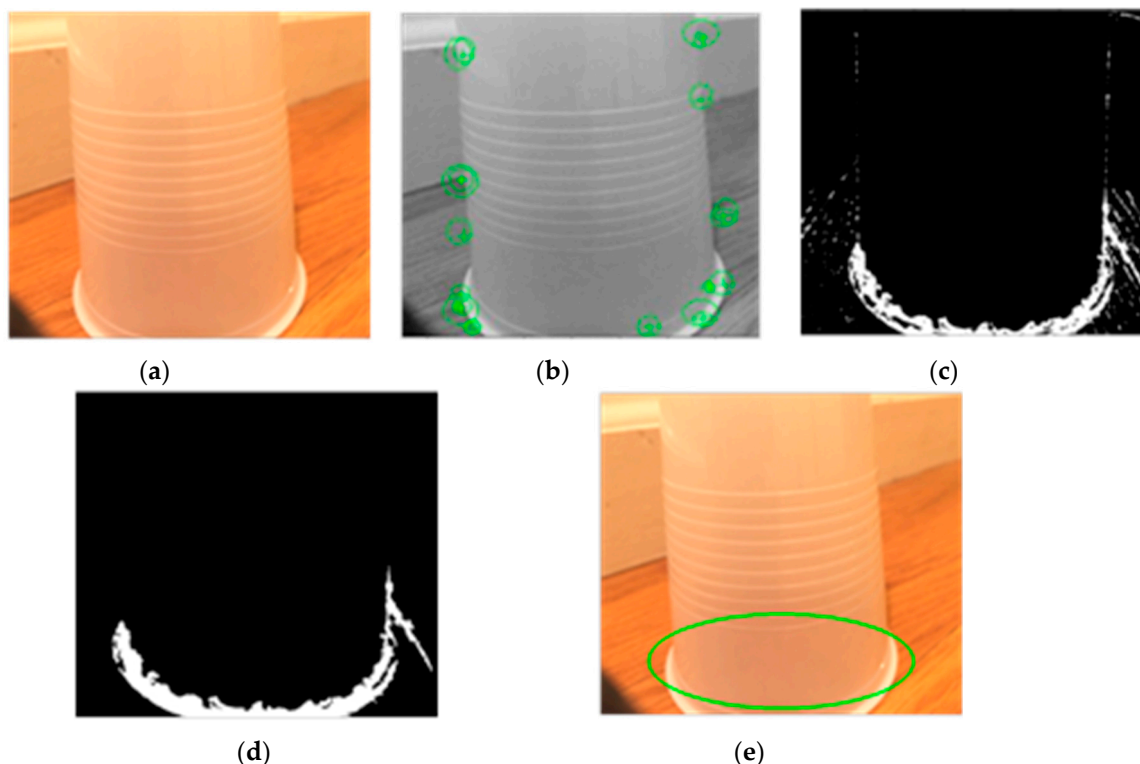
### 6.3. Infrared and Ultrasonic Sensors

At the initial position ( $t_0$ ), no object is detected, and the infrared distance measuring sensors produce invalid values and no distance is recorded for the ultrasonic sensor. At time  $t_5$ , a front object is detected by all of the sensors within a distance of 18 cm. At a time of 41 s, a front right object is detected within a distance of 16 cm from the right infrared sensor and 12 cm from the ultrasonic sensor. At a time of 72 s, a left front object is detected through the left infrared sensor within a distance of 6.5 cm. At time  $t_{93}$ , the robot detects another left front object as the left infrared sensor returns a distance of 10 cm and an ultrasonic sensor return distance of 13 cm, while the right infrared sensor returns an invalid measurement. The robot will continue moving forward while detecting objects and avoiding collisions along its path.

### 6.4. Camera Module

Object detection flow by the camera is depicted in Figure 16. A series of observations has been considered to determine the stability of detecting objects with a moving camera. The performance of object detection by the camera was evaluated by using two videos, with each having a different number of frames and objects. The final evaluation results are presented in Table 4, where a true positive is the number of accurate detections and a false positive is the number of inaccurate detections. The Detection Rate represents the percentage of correct detections.

Table 5 provides a comparison between the proposed method and state-of-art methods. The only method that displays similar asymptotic complexing to the proposed method is AFP, which performs poorly with regards to the correspondence of narrow passages.



**Figure 16.** (a) A selected frame from the reference video; (b) Feature points detected; (c) Resulting binary masks from the segmented image (where abandoned objects appear); (d) The resulting binary image is cleaned with morphological filtering to clean up the mask image and remove points that do not represent the object; (e) The result of the detected object.



**Table 4.** The final evaluation results of object detection by the camera.

Video	Number of Frames	Number of Objects	Number of Detected Objects	True Positive	False Positive	Detection Rate
1	27	19	17	16	1	84.21%
2	76	58	51	46	5	79.31%

**Table 5.** Comparison between the proposed method and state-of-art methods.

Method	Modules	Time Complexity	Effectiveness	Comments
<i>FGM</i>	Ultrasonic sensor, laser Sensors, Camera, Processor.	Obstacle avoiding using “FGM” is completed in three main steps $O(n(\log n)^2)$ .	Able to select shortest path and avoid symmetric obstacles.	Fast processor is needed. Fails for U shaped obstacles.
<i>APF</i>	Ultrasonic sensor, IR distance Sensors, Microcontroller.	Less time required as it selects shorter path. $O(n \log n)$ .	Difficult to use in real-time application.	Unable to detect symmetric obstacles.
<i>VFH</i>	Ultrasonic sensor, High memory, Processor.	High costs on computation and histogram generation.	More resources required for calculation.	Fast processor is needed for modeling. Able to select shortest path.
<i>BA</i>	Ultrasonic sensor, IR distance Sensors, Microcontroller.	Its unidirectional obstacle detection increases the travel time.	May take robot away from goal.	The selected path is not the shortest. No obstacle avoidance.
<i>Proposed Method</i>	IR sensors, Ultrasonic sensors, Microcontroller.	Low cost sensors and Low computational power $O(n \log n)$ .	Real time obstacle detection.	Easy to tune, and able to detect Symmetric and U shaped obstacles.

### 6.5. Summary

In this paper, a set of experiments were performed on the FEZ Cerbot mobile robot in a lab environment. The robot was equipped with multiple sensors to create an alert if an object is detected while moving along its path. The optimal motion planning is introduced to optimize the energy consumption during motions where an algorithm has been used to create a collision-free path. The path planner saves energy by computing the shortest path (least-cost path) from a given position to the target position. The frame work has an asymptotic complexity of  $O(n \log n)$  for energy consumption, and the time and distance traveled, where  $n$  is the number of obstacles. The experimental results demonstrated the effectiveness of the proposed model in terms of distance and time traveled, energy consumption, and accuracy. The proposed model does not require prior information of the environment, as the decision is based on the current percepts captured by the sensors. A demonstration of the simple and complex experiments is available in [17].

Cameras mounted on mobile platforms, as observed in most aerial surveillance or reconnaissance, tend to capture unwanted vibrations induced by the mechanical parts of the platform. A straightforward approach to overcome this problem is to eliminate the motion induced in the camera through the aerial platform, which is also known as ego-motion compensation in computer vision literature. Once the motion of the moving platform is compensated, the next task is to identify Regions of Interest (ROIs) from the video, where all of the moving objects fall under the umbrella of ROI.

In this paper, we built a separate block diagram for object detection flow by a camera, as shown in Figure 2. A series of observations have been considered to determine the stability of detecting objects with the moving camera. The performance of object detection by the camera was evaluated by using two videos, with each having a different number of frames and objects. The final evaluation results are presented in Table 4.

The starting position and goal position are predefined, as stated in Section 5.2. The experiments were carried out to find an optimal collision path in simple and complex dynamic environments, where the robot started from a given initial position and needed to travel toward a predefined goal position using the shortest path.

The proposed model deploys a searching algorithm to find the shortest path from a given point to the target. The algorithm has been widely used in path finding for its reduced search space; it first searches, collecting an input, and then evaluates a number of possible paths and returns the least-cost path from a given source to the destination [15]. As the robot follows the least-cost path, it maintains a sorted queue of alternate paths along the way, to abandon a higher cost path segment at any point in time, and use the lower-cost segment instead. This procedure continues until the target is reached.

When the shortest path from the initial position to the target has been determined, the proposed collision avoidance model performs two processes: edge detection and obstacle detection.

Although there are other approaches introduced in the literature including deep learning approaches [18,19] and sensor-based analyses [20], these works are not relevant to our work as they use a vision-based model. They use a set of indoor depth data-sets for training, where the ground-truth output is instructed by a human operator. This dataset is used to train the proposed model. In our work, the main goal is to build a framework for collision-free path planning in a mobile robot, as presented in Section 4. Our model has been tested on a real robot with multiple scenarios ranging from simple to complex environments. A reliable path planning algorithm should feature a low computational complexity, short traveling distance, and smooth trajectory, while avoiding collisions. Moreover, it must guarantee the safety in the dynamic environment, while traversing toward the goal with a low cost and short travel time. The collision-free path model used in this work checks for all reachable solutions to the destination and then follows the optimal one (the shortest path). The algorithm is exploring the state space by generating pointers of already-explored states, in order to avoid expanding paths that are already expensive. Therefore, the overall performance of our model has been evaluated from different points of view (based on distance traveled, travel time, and energy cost).

Table 5 provides a comparison between the proposed method and state-of-art methods. The only method displays similar asymptotic complexing to the proposed method is AFP, which performs poorly with regards to the correspondence of narrow passages.

This work mainly focuses on finding the optimal path from a given position to the target position. Our goal is to build a framework for collision-free path planning in a mobile robot, as presented in Section 4 (Description of Proposed Approach). Our model has been tested on a real robot with multiple scenarios ranging from simple to complex environments. The proposed model does not require prior information of the environment, as the decision is based on the current percepts captured by the sensors.

In this paper, we did not address the sensor fusion model as we already presented a collision-free mobile robot navigation based on the fuzzy logic fusion model in [21,22]. Eight distance sensors and a range finder camera are used for the collision avoidance approach, where three ground sensors are used for the line or path following approach. The proposed methodology, which includes the collision avoidance based on the fuzzy logic fusion model and line following robot, has been implemented and tested through simulations and real time experiments. Various scenarios have been presented with static and dynamic obstacles using one robot and two robots, while avoiding obstacles of different shapes and sizes.

## 7. Conclusions

The field of collision-free path planning algorithms for a mobile robot in dynamic environments is one of the well-studied problems that has a large scope. In this work, an integrated framework is presented to produce optimal motion planning for a wheeled mobile robot. The framework has an asymptotic complexity of  $O(n \log n)$  for energy consumption, and the time and distance traveled, where  $n$  is the number of obstacles. The path planner of our model is utilized to generate the shortest path from a given initial position to the destination. The experimental results demonstrated the effectiveness of the proposed model in terms of the distance and time traveled, energy consumption, and accuracy. The proposed model does not require prior information of the environment, as the decision is based on the current percepts captured by the sensors.

The proposed model presented in this work can be extended in many promising directions. In order to extend our study in the future, the proposed method will be applied to resolve several problems such as multiple robots in dynamic environments for energy minimization, to verify the validity and practicability of our method in unknown environments. Furthermore, several methods have been reported lately which combined the collision-free path planning algorithms with neural network and other machine learning approaches. Considering a combination of these approaches with the proposed model is an important topic that is worthy of investigation.

**Acknowledgments:** The authors acknowledge the support of the University of Bridgeport providing the necessary hardware platform and computational resources to complete this research.

**Author Contributions:** The work has been primarily conducted by Abrar Alajlan under the supervision of Khaled Elleithy. Abrar Alajlan wrote the manuscript. Extensive discussions about the algorithms and techniques presented in this paper were carried between Abrar Alajlan, Khaled Elleithy, Marwah Almasri, and Tarek Sobh over the past year.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Abiyev, R.; Ibrahim, D.; Erin, B. Navigation of mobile robots in the presence of obstacles. *Adv. Eng. Softw.* **2010**, *41*, 1179–1186. [[CrossRef](#)]
2. Ullah, I.; Ullah, Q.; Ullah, F.; Seoyong, S. Integrated collision avoidance and tracking system for mobile robot. In Proceedings of the IEEE Robotics and Artificial Intelligence (ICRAI), Islamabad, Pakistan, 22–23 October 2012; pp. 61–67.
3. Fox, D.; Burgard, W.; Thrun, S. A hybrid collision avoidance method for mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Lueven, Belgium, 16–20 May 1998; pp. 1238–1243.
4. Marco, B.; Gabriele, B.; Cacao, M.; Lapiere, L. A collision avoidance algorithm based on the virtual target approach for cooperative unmanned surface vehicles. In Proceedings of the IEEE 22nd Mediterranean Conference on Control and Automation, Palermo, Italy, 16–19 June 2014; pp. 746–751.
5. Duleba, I.; Sasiadek, J.Z. Nonholonomic motion planning based on newton algorithm with energy optimization. *IEEE Trans. Control Syst. Technol.* **2003**, *11*, 355–363. [[CrossRef](#)]
6. Zeng, L.; Bone, G.M. Mobile Robot Collision Avoidance in Human Environments. *Int. J. Adv. Robot. Syst.* **2013**, *10*. [[CrossRef](#)]
7. Yufka, A.; Parlaktuna, O. Performance comparison of bug algorithms for mobile robots. In Proceedings of the 5th International Advanced Technologies Symposium, Karabuk, Turkey, 13–15 May 2009; pp. 61–65.
8. Chen, K.H.; Tsai, W.H. Vision-based obstacle detection and avoidance for autonomous land vehicle navigation in outdoor roads. *Autom. Constr.* **2010**, *10*, 1–25. [[CrossRef](#)]
9. Zhu, Y.; Zhang, T.; Song, J.; Li, X. A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge. *Knowl. Based Syst.* **2012**, *27*, 302–313. [[CrossRef](#)]
10. Sezer, V.; Gokasan, M. A novel obstacle avoidance algorithm: “Follow the Gap Method”. *Robot. Autom. Syst.* **2012**, *60*, 1123–1134. [[CrossRef](#)]
11. Oroko, J.; Ikua, B. Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: A Review. In Proceedings of the 2012 Mechanical Engineering Conference on Sustainable Research and Innovation, Juja, Kenya, 3–4 May 2012; Volume 4.
12. Babinec, A.; Dekan, M.; Ducho, F.; Vitko, A. Modifications of VFH navigation methods for mobile robots. *Procedia Eng.* **2012**, *10*, 10–14. [[CrossRef](#)]
13. Borenstein, J.; Koren, Y. The vector field histogram—Fast obstacle avoidance for mobile robots. *IEEE J. Robot. Autom.* **1991**, *7*, 278–288. [[CrossRef](#)]
14. Yan, Z.; Zhao, Y.; Hou, S.; Zhang, H.; Zheng, Y. Obstacle avoidance for unmanned undersea vehicle in unknown unstructured environment. *Math. Probl. Eng.* **2013**, *2013*, 841376. [[CrossRef](#)]
15. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [[CrossRef](#)]

16. Hu, H.; Brady, M.; Probert, P. Coping with uncertainty in control and planning for a mobile robot. In Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka, Japan, 3–5 November 1991; pp. 1025–1030.
17. Alajlan, A.; Elleithy, K. Autonomous Mobile Robot Navigation for Planning Collision Free Path in Static and Dynamic Environments, Video. Available online: [https://www.dropbox.com/s/zra8w2ib7afyf21/Abrar\\_Alajlan\\_Demo.mp4?dl=0](https://www.dropbox.com/s/zra8w2ib7afyf21/Abrar_Alajlan_Demo.mp4?dl=0) (accessed on 29 March 2017).
18. Tai, L.; Li, S.; Liu, M. A Deep-network Solution towards Model-less Obstacle Avoidance. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016.
19. Mirowski, P.; Pascanu, R.; Viola, F.; Soyer, H.; Ballard, A.; Banino, A.; Denil, M.; Goroshin, R.; Sifre, L.; Kavukcuoglu, K.; et al. Learning to navigate in complex environments. *arXiv* **2016**, arXiv:1611.03673.
20. Liu, M. Robotic Online Path Planning on Point Cloud. *IEEE Trans. Cybern.* **2016**, *46*, 1217–1228. [[CrossRef](#)] [[PubMed](#)]
21. Almasri, M.; Elleithy, K.; Alajlan, A. Sensor Fusion Based Model for Collision Free Mobile Robot Navigation. *Sensors* **2016**, *16*. [[CrossRef](#)] [[PubMed](#)]
22. Almasri, M.; Alajlan, A.M.; Elleithy, K. Trajectory Planning and Collision Avoidance Algorithm for Mobile Robotics System. *IEEE Sens. J.* **2016**, *16*, 5021–5028. [[CrossRef](#)]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).