Journal of Theoretical and Applied Information Technology
29th February 2020. Vol.98. No 04

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ROBOT NAVIGATION SYSTEM USING LASER AND A MONOCULAR CAMERA

TAMER ABUKHALIL [1,*], MALEK ALKSASBEH [1], BASSAM ALQARALLEH [1] AND ANAS ABUKARAKI [1]

[1] Department of Computer Science, Alhussien Bin Talal University Ma'an, Jordan
* Correspondence: tamer.abukhalil@ahu.edu.jo;
Received: date; Accepted: date; Published: date

ABSTRACT

The key contribution in this research work is introducing a technique that can be used to avoid unknown objects to a mobile robot. Most vision-based obstacle avoidance systems tend to have high computational complexity in order to compensate for accuracy. The proposed method tries to overcome that using pattern recognition of two laser pointers in the view of a single camera. This vision combination along with the program that runs on a base station is designed as a module that detects objects around the robot. Such obstacles are detected by calculating the intensity of the red laser points found on each frame being captured by the camera in real-time. Distance and angle to the objects is measured using Lagrange interpolation formula applied separately to each laser projection in the framed image. A map is created to show the robot's actual distances versus estimated ones as the robot keeps track of objects which are in the camera's view. The algorithm successfully manages to avoid obstacles as shown in the experiments. The effectiveness of the proposed system is evaluated by deploying the robot and performing a simple navigation task. The results show that the concept algorithm along with the hardware module is able to utilize the monocular vision with classification error of 3.63%.
Keywords: *Monocular Camera, Obstacle Avoidance, Distance Measure, Lagrange Formula, Mobile Robots*

1. INTRODUCTION

A very important aspect in the problem of navigation is obstacle avoidance. Sensor-based approaches are typical research solutions to robotic obstacle avoidance and are well-investigated [1, 2]. In the sensor-based navigation work carried out in the literature, most obstacle avoidance applications are accomplished using ultrasonic, infrared, or laser sensors. The overall sensing system is responsible to scan and detect obstacles. Once an obstacle is detected in the robot's path, the control is taken by the vision system at which some calculations have to be performed. The control is then given back to the robot's vision-based processing when the obstacles are no longer in sight. Despite that fact it is more common to use ultrasonic sensors in such case, it is interesting to have a backup vision-based avoidance should such sensors fail to operate. Also, humanoid robots definitely rely in their vision and navigation system on stereo cameras rather than regular ultrasonic or sonar sensing units. Humanoids and advanced robotics focus on imitating human behavior where humans use vision for obstacle avoidance when navigating. Therefore applying such capabilities in a robotic system is worthy.

When it comes to vision-based depth perception, there are three categories of approaches that can be used to extract features and depth perception from a two-dimensional frame. They can be categorized as monocular vision [3, 4], stereo cameras [5], and motion parallax [6]. Many researchers have studied stereo vision and this category is considered the most common area. However, in order to have simplified hardware configuration, mono cameras has caught researchers' attention especially when designing decentralized robotic systems. Depth obtained from motion or optical flow in buffered images is called motion parallax [7]. This approach basically works by discovering correspondence between specific pixels in several buffered frames divided over Cartesian space or time which is called optical flow of images. When an accurate correspondence of a particular pattern is found, this method successfully obtains depth estimates very precisely. However, it is computationally expensive to search for such

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pixel patterns, this process can dramatically slow down the robot's overall vision performance.

In this paper, we present an integrated vision-based algorithm that runs on a mobile robot to allow it to simultaneously navigate and avoid stationary as well as moving obstacles using monocular camera images. Despite the fact that simultaneous localization and mapping (SLAM) is partially implemented in the proposed system, the key contribution of this work is to develop a robotic vision-based obstacle system that uses simpler and lower cost hardware components. The vital components are basically, a single camera and a couple of laser emitting devices mounted in the front so that their emitted light can be seen by the camera. In the experiment, the obstacle-avoidance capability successfully detects and avoids objects even if they are moving, despite the computational limitations of hardware mounted on the robot. The robot communicates with a central station that runs a pattern recognition program.

2. LITERATURE REVIEW AND CONTRIBUTION

Many research articles have been introduced in machine vision for mobile robotic systems. Scott Lenser el. al.[8] designed a vision algorithm to detect obstacles which allows partial identification of objects. The objective is to enable the robot to selectively avoid unknown obstacles while recognizing the desired objects. However, their main contribution is the introduction of a vision hardware mounted on the robot that can adapt to different background colors. The designated electronics allowed the robot to follow white lines on the floor while avoiding a white wall. However, unlike our approach, their technique requires that the floor must have a uniform color.

Other approaches such as [9, 10] require additional vision onboard equipment that are dedicated to depth perception. Such application can be found in automobiles, more specifically; head-up displays which can be used in automatic driving (auto-pilot) and the overall awareness of the surroundings[11]. Some other systems require a wireless communication to a centralized computer. In robotic systems, however, many of the related vision approaches are based on detecting objects according to their heights in the surrounding environment.

An omni-directional camera is used in the OMNI RoboCup [12] which is a team of mall-size robots. Each robot sees its surroundings by segmenting the image based on background color to find an open way to go. However, their algorithm may get confused with big obstacles and sometimes is not able to handle areas consisting of multiple or overlapping colors. Ulrich and Nourbakhsh [13] use a gray scale histogram to segment images captured by an omni camera into partitioned floor and obstacle regions. They also use transformation operation per pixel which consequently increases the complexity of the algorithm.

Reactive approaches can also be used to visualize objects. For example, the optical flow method used in [14] can estimate the relative distance to an object in a quadrotor robot by detecting changes in the size of the object in the image. However, this method is limited to side-view cameras, because the optical flow of size changes of obstacles which appear in the sides is much slower than that of frontal ones. Next to optical flow approaches, many researchers introduced fusion algorithms as an attempt to enhance the obstacle detection accuracy. For example, C.S. Dima et al.[15] introduced a fusion algorithm to classify objects in images by developing a technique that combines color and infrared images along with relative measurements from proximity rangefinder sensors. Their experiment has been depicted in images and was conducted outdoors in autonomous off-road navigation [16]. The results showed a high success rate, however their algorithm is highly dependable on supervised learning and labeling data. On the other hand, A. Vatavu et al. [17] proposed a method that take advantage of objects associations in order to detect obstacles using multiple sensors. The results showed a high detection rate with minimal errors. Other types of fusion algorithms have also been implemented to detect obstacles within another type of sensors [18, 19]. In some other proposals, GP Stein et. al. [20] introduced a technique to detect relative size change in the image frame of approaching objects. Lin and Song [21] proposed a method using an imitation learning algorithm, in which multiple optical flow cues and visual features are used to detect moving objects in the image.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

When the robot is fitted with two cameras at the front, stereo vision methods can be used to obtain a depth perception [22]. In [23], a MAV (Micro Arial Vehicle) robot is wirelessly connected to a laptop which runs a SLAM (Simultaneous localizing and Mapping) algorithm based on data generated by ultrasonic sensors. The same scenario is also presented by Blösch et al. [24] however, in their work; authors demonstrated visual localization using a single camera to perform a SLAM procedure in structured environment rather than relatively unknown surroundings. Because images are inexpensive to acquire, and unlike classical two-view stereo methods, multiple images of the environment can be used to generate more accurate depth maps. For instance, Newcombe et al. [25] proposed in their DTAM approach, the idea of adding up hundreds of images before generating a depth map. Their approach however, only tried to avoid collisions and does not necessary detected the exact distances. The collision avoidance technique simply detects the differences in object size in the various images

The outdoor MAV robot in [26] is fitted with a joint combination of a mono camera for obstacle detection and a GPS system for position control. When only a single camera is available (as an attempt to reduce costs), stereo vision can also be generated using several consecutive images taken from the same camera at the moment when the camera is moving. This is a part of motion parallax operation as discussed before. However, the challenge of working with cameras is that the system has to deal with a high level of noise that is typically associated with data obtained from stereo vision. Moreover, stereo-vision detection usually has a limited and narrow field of view which may degrade detection accuracy as the distance to the obstacle increases. Indoor MAV was investigated in [27], H. Madokoro et al. demonstrated an indoor SLAM integrated on a MAV using a monocular camera without the use of central station; however their system did not utilize object detection and classification. In general, the use of cameras in detecting obstacles may result in substantial increase of computation load. Just replying on a mono or stereo cameras is not feasible. There will always be a tradeoff between the accuracy of distances being detected and the computational time. The suggested proposal here tries to overcome such drawback by utilizing laser pointers that illuminate pixels of specific range in the view of the camera. The computational load will then be dramatically reduced because only smaller amount of pixels are analyzed by the software module.

The proposed Module uses a high-level environment-independent algorithm for vision-based navigation. The proposed methodology excels on previous obstacle avoidance techniques because it is able to handle a different kind of environments. Particularly, the technique integrates two popular distance measurement algorithms namely langrage and approximation formulas. Field environments in which the robot was tested include spaces in which the ground may have variety of visual texture such in outdoor scenarios.

This paper is organized as follows; section 3 brings into discussion the vision system and its components. More details on the vision algorithms are also introduced. Section 4 presents the experimental evaluation and a sophisticated description on the hardware of the designated robot. And finally, section 5 and 6 draw out a conclusion and future ideas.

3. VISION SYSTEM

The idea is inspired by simplified pattern recognition of a particular region of pixels. In order to make the robot perceive and understand its distance from surroundings, the vision problem can be formulated as a depth estimation procedure performed over frames of the input image.

The algorithm uses the well-known Lagrange interpolation formula to predict the distance to an obstacle. The prediction is based on the actual buffered images from the camera and the embedded laser dots. In the proposed system there are actually two laser dots that are positioned at the right and left center of the captured images. Consequently, there are two depth perception ways that in turn will read the left and right distances. The robot uses these distances to decide which way to go. If the two distances are equal, the robot simply backs up and turns to one side in random. The algorithm works by going through all the pixels in the buffered image. The robot interacts wirelessly with a centralized computer where the algorithm is executing.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The algorithm begins by going through each pixel and a number of steps are repeated infinitely as long as the robot is moving. Scanning for the X,Y pixels is initialized after 40% in the Y-coordinate where the special RGB pattern is most likely to be, whereas, the initial X-coordinate is 0. Within the body of this loop the algorithm checks if the current pixel is in the filter's range by analyzing the pattern. Once the algorithm finds the special RBG pattern indicating the left laser dot, then it sets the current location's X (lo_x) value to the current X when the left pixel detected is less than the left laser dot location. If the right pixel detected is more than the right laser dot location set the current location's X (hi_x) value to the current X. The current location will be later sent to Lagrange procedure in order to determine the distance. Algorithm 1 is illustrates the above steps

First of all, the distance threshold at which the robot should not move any further is set at 0.3 meters. Hence, the threshold is set at 30 cm. At this point the robot turns to a particular direction based on calculations as described in the following algorithms.

obstacleDistance ← 0.3 (threshold);

Algorithm_1 *processImage(image)*

1. *w ← image_Width;*
2. *h ← image_Height;*
3. *current_loX ← ∞*
4. *current_hiX ← −∞*
5. *for each y starting from 0.4*h to h*
6. *Begin_for*
7. *for each x starting from 0 to w*
8. *Begin_for*
9. *if In Color Range(x,y) {*
10. *if (current_loX > x) {*
11. *if ((x+1 < w) and (y-1 > 0) and(y+1 < h) and In_Color_Range(x+1, y) and*
12. *In Color Range(x+1, y-1) and In_Color_Range(x+1, y+1))*
13. *{*
14. *Current_loX ← x;*
15. *x ← x+1;*
16. *} }*
17. *if (current_hiX < x)*
18. *if ((x-1 > 0) and (y-1 > 0) and (y+1 < h) and In_Color_Range(x-1, y) and In Color Range(x-1, y-1) and In_Color_Range(x-1, y+1))*
19. *Current_hiX ← x; }*
20. *End for*
21. *End_for*
22. *Returns current loX and current hiX*
23. *End procedure*

A small microcontroller is responsible to fetch images off the camera and together with the onboard electronics, the robot's main chip communicates with a program running on a central desktop computer. The software is embedded with a bunch of routines. The second algorithm searches for the laser dots in the image after it is converted to frames in a buffer. This is accomplished by converting the pixel into its RGB composition. This procedure is carried out by the function *In_Color_Range*. Since each Red , Green or blue value is in the range of [0 .. 255] , the *In_Color_Range* function checks if the RGB values lie within a specific range and returns "True" if the condition is satisfied. The function is shown in algorithm 2 as follows:

Algorithm 2*: In_Color_Range(x,y)*

1. *Color ← getRGB(x,y)*
2. *r ← color.red;*
3. *g ← color.green;*
4. *b ← color.blue;*
5. *if (r > 240) and (120 < g < 230) and (120 < b < 230))*
6. *return true*

3.1 Lagrange Formula

When the required pixels that indicate laser dots on the image are found, Lagrange interpolation polynomial can be used to generate polynomial functions and to predict the most fitted curve of values that thereafter can be used for numerical analysis. The interpolating polynomial is embedded within the program and it is executed in real-time. The integrated polynomial should be of the least square degree

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in order to maximize the accuracy of the fitting curve that is exists between the data points. Lagrange estimation method is applied particularly on pixels of interest indicating the laser dots (X-coordinates) in the famed image; therefore, Y-values are computed when the training data follows a uniform consistent arrangement. The interpolation function $f(x)$, is given in the following equation

$$f(x) = L_{n,0}(x)\, y(x_0) + L_{n,1}(x)\, y(x_1) + \cdots + L_{n,n}(x)\, y(x_n) = \sum_{k=0}^{n} L_{n,k}(x)\, y(x_k) \quad (1)$$

Where $L_{n,k}(x)$ are the LaGrange basis polynomials defined by $L_{n,k}(x) = \prod_{i=0}^{n} \frac{x-x_i}{x_k-x_i}$ and $L_{n,k}(x_i) = 0$, $L_{n,k}(x_k) = 1$

In the LaGrange equation (1), $f(x)$ represents the distance in meters for n points. X-coordinate is the position of the pixels of interest within the frame. The interpolation of X against the reliant variable Y is generated using the formula in real-time. Algorithm 3 uses these values given in the table initially and tries to predict upcoming real distances based on the position values of the intended pixel in the 2-D world. Some values will be similar to the values given in the table. The training set is composed of n records denoted by $(x_1, y_1), \ldots, (x_n, y_n)$. It is supposed that Y is the distance in meters, these values specify the vector of points $\vec{x}_i$ by performing the mathematical prediction code found in line 6 in algorithm (3). The formula tries to predict values which form a line that fits the collection of $\vec{x}_i$ vectors based on values of $y_i's$ out of the Y coordination. Table 1 shows the learning set for each pixel's X locations and the corresponding Y values which represent distances in meters. The pixel position will be fed to Lagrange interpolation function and the output will be the predicted distances.

Table 1: learning set (pixel values and corresponding distances)

Position (pos)	Value (meters) Val
0	0.06
1	0.065
5	0.067
10	0.07
30	0.075
50	0.08
70	0.15
96	0.28
128	0.56
135	0.84
143	1.12
150	1.4

The system proceeds to predict the relative distances to obstacles using Lagrange formula and converts the distance which is initially computed as pixel distance to real distance in meters. Once the distances of both lasers are calculated the algorithm decides which direction to turn to depending on those two values, the bigger the difference between the two the smaller the rotation angle is. Algorithm 3 shows the Lagrange interpolating procedure.

Algorithm 3: lagrange Interpolating (desiredPos, matrix pos,val) returns retVal

1. *retVal ← 0;*
2. *for each i from 0 to pos_length*
3. *Begin for*
4. *weight ← 1;*
   for each j from 0 to pos length
   {
5. *Begin_for*
   if (j ≠ i) {
   *weight ← weight *(desiredPos – pos(j)) / (pos(i) – pos(j));*
   }
6. *j ← j +1;*
7. *End for*
8. *retVal ← retval+(weight * val(i));*
9. *i ← i + 1*
10. *End for*
11. *return retVal*

4. EVALUATION

The system is tested on robots that are configured with latest hardware devices and parts. The following subsections show the hardware configuration of one particular robot as well as a discussion of a simple task and the relative results. The robot is actually controlled by commands being sent from the base station

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

whenever it encounters an obstacle. Any other simple subtasks are performed by a small routine running on the robot's circuit board.

4.1 .Setup

A specific program that runs on a small robot is designed to tolerate sensor failure by adapting redundant obstacle detection schemes. From the hardware point of view, the robot is controlled by a circuit board, which consists of a 1000 MIPS 500MHz processor of type PIC33EP digital microcontroller, which operates at a clock speed of 60.07MHz. The communication module attached to the board is an XBee S1802.15.4 1 serial modem that operates at 1 *mw* (mili Watts). One of the key requirements in the hardware design is to provide a steady clock for the high-speed serial communications with negligible timing errors, a 6.38 MHz clock was chosen, and a 7V-24V DC voltage regulator, and digital/analogue of headers for peripheral ultrasonic and sonar sensors.

The robot is also equipped with a single monocular camera of type Omnivision OV9655 1.3 megapixel 160x128 to 1280x1024 resolutions [28]. Two laser pointers emit a red laser light that reflects on obstacles in front of the robot as it wanders around as shown in figure 1. Therefore, these two laser dots will always be in the sight of the robot's camera. The output of the camera along with the reflection of laser lights are fed as an input to the higher level control algorithm running on a computer. The communication is achieved using wireless LAN at 30 fps (frames per second). The robot and interacts wirelessly within 802.11b/g Wi-Fi interface. The robot is also equipped with a 2-DoF (Degree of freedom) gripper at the front. The gripper can be used in some other tasks as well. The software module runs on a computer powered by Intel Core i7-3122 CPU. The wireless communication channel allows multiple robots to communicate with each other and the base station should there be more than a single agent.

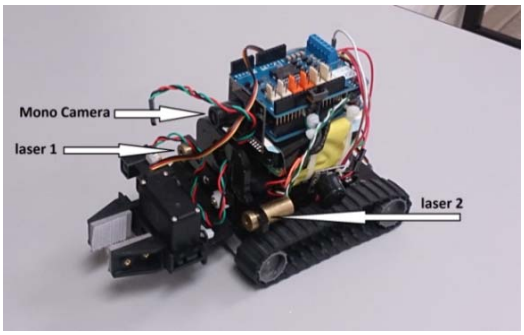

Figure 1: robot with mono camera and two lasers

Figure 2: Laser dots and Distance being measured

4.2 Obstacle Avoidance Experiments

Figure 2 shows the real distance being calculated and the action being taken by the robots as it wanders around the room. The figure also shows the laser dots reflected on the wall. In terms of number of obstacles avoided, the success ratio is 92%. Table 2 shows the experimental results. In order to make the robot learn estimating distances to obstacles in an indoor scene, multiple runs have been conducted. The robot has been tested 25 times through this experiment course. Theoretically, the proposed setup should work fine; however, there are 2 failure cases where the robot was not able to avoid the obstacles. Such rare failures could be caused by the high proximity of the camera to the ground and bad lighting that does not provide enough color separation (contrast) in the captured image. The speed of the robot is also

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

vital. Another factor is that the wireless communication and sending video frames to the main computer has to catch up with the speed of the robot. The bad lighting circumstance has been exploited to test the performance of the image processing technique in these kinds of environments.

Table 2: Results of total 23 runs

	Total	Success	Failure	Ratio
Run	25	23	2	92%
Objects	14	13	1	92.8%

Figure 3 shows the trajectories of three consecutive runs. The system is tested thoroughly in an environment which contains various different color objects and shapes. The robot wanders around and successfully avoids obstacles as long as the objects are high enough to be seen by the robot. Obstacles that appeared suddenly within close range were also avoided by the robot. Time delays associated with image processing were handled by using relatively fast communication provided by the hardware and the procedures as discussed in section 2.

The behavior of the robot as can be seen in the figure tends to have different paths in each run but the overall movement is almost the same. While navigating the obstacles that happen not to be in the view of the camera are detected and avoided using two sonar sensors attached to the sides and at the rear. Such situation would occur when the robot is backing up. A set of waypoints are generated in real-time as the robot wanders around. At each point the distance between the robot and the obstacle is measured so that the differences and margin of error can be calculated and compared with the estimated distances calculated by the laser scan along with the mounted camera, as depicted in figure 4. The waypoints are initially created virtually. To show a graphical motion view of the robot within the experiment area, a map is generated using the LabVIEW [29] virtual platform to display the robot's maneuvers. There has been much emphasis in obtaining a uniform and consistent behavior from the robot in the multiple runs. The primary solution is directly related with the very design of the robot.

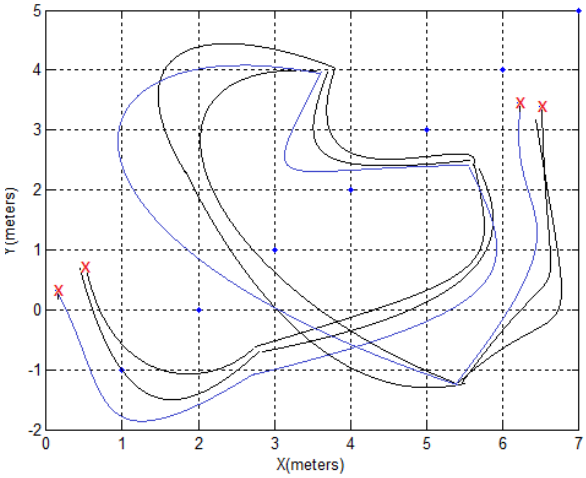

Figure 3: The navigated path of the robot

A rigid frame is the evident factor which saves a lot of effort in maintaining consistent paths and the overall structure integrity. The application of threads instead of rubber wheels also helped in obtaining high level of precision while turning and maneuvering. For obstacles that may appear in the back or on the sides of the robot, a set of ultra-sonic sensors have been used. The onboard computer receives data from sensors and manages the motion.

As the robot moves, the current Cartesian location at which it detects an obstacle is stored. At the same moment, the current location of the robot is recorded as well. A line is drawn from the previous waypoint to the current one in order to show the path being travelled. Once the robot gets to the goal, the final location is registered on the map. Out of the 950 waypoints that have appeared in the path where the robot is navigating, only 33 were miscalculated as shown in figure 4 with red notes (dots). The relationship between actual and estimated distances is compared using the blue (notes) crosses as seen in the figure. Five out of the 33 waypoints appear to have been mistaken by unexpected pixel position values in the training data shown in table 1, while the rest of the errors are resulted by bad environment conditions such as bad lighting causing a classification error of 4.7%.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

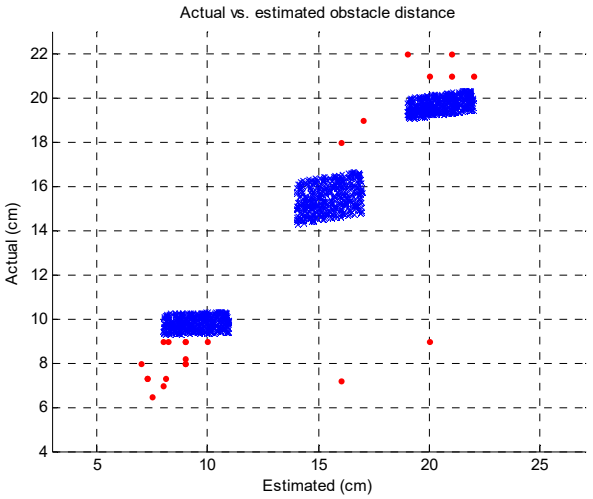

Figure 4: Actual vs. estimated distances

The 5 values of invalid pixel positions have been corrected, resulting in a classification error of 3.63% where the error is the square root of mean differences between the estimated and actual values.

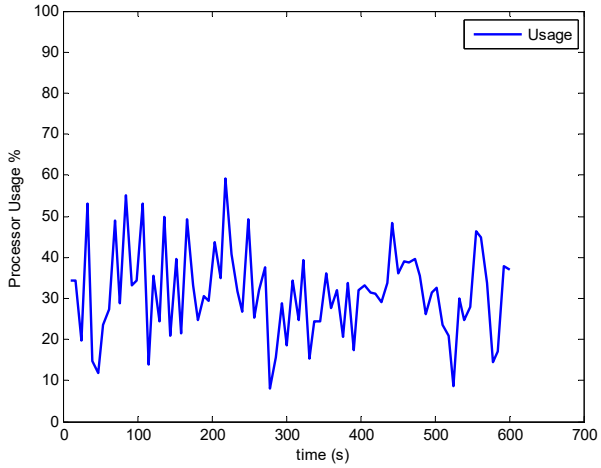

Figure 5: Processor usage in run time

Validating the effectiveness of the system has been shown and investigated in the figures above. Additionally, another important aspect of the presented work that should be addressed is the analysis which evaluates the efficiency of the system from the processor perspective. It is important to mention that even though the avoidance and recognition program runs on a base station, the onboard camera module is responsible to capture images in real-time. Figure 5 shows the percentage of usage for the onboard microcontroller during running time in the experiment.

Typically, navigation is not the only task that is assigned to the mobile robot. For that reason, power saving is of great concern in this scenario. Most power is consumed by the servo motors attached on the wheels. However, using the proposed approach greatly enhances power reservation. Having in mind that the mobile agent is running using batteries, the system should be evaluated in terms of power consumption. This is one of the objectives that are going to be discussed in future research.

More specifically, the process that utilizes LaGrange equation does not require supervised learning unlike the fusion-based approaches discussed before. Table 3 brings into discussion the major limitations concerning the current approaches. Consequently, the main contribution behind this work will be more understood. The table also projects the position of the proposed work amongst other approaches. Moreover, the proposed approach can be combined with other type of cues. As the matter of fact, the given approach can be extended using multiple cameras so that consecutive images can be used to determine the relative object size and shape.

5. CONCLUSION

In this paper, a new approach to obstacle avoidance for a robotic agent that has only a single monocular camera has been introduced. Each of the current proposals in vision-based navigation has its own limitations in terms of efficiency and adoptability. This work brings up a total new concept in robot navigation and avoidance context.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Comparison And Key Features Of Current Approaches.

Approach	Method	Remarks	References
Monocular camera	-Partial identification (perspective) -Segmenting the image	-Mainly uses size expansion cues (not good for frontal objects). - Works perfectly in known environments. -cannot detect distances, it is just used to detect approaching obstacles.	[3, 4, 8, 9, 11-13]
Stereo vision	Depth perception using two images.	-Need a great amount of resolution. - can be combined with optical flows. Especially in obstacle that appear in the front.	[5, 22-26]
Optical flow	-Buffered frames. -Reactive methods	- Difficult to be used to avoid frontal obstacles -Mainly relies on motion flow that requires high resolution in the camera. - Sometimes its combined with stereo vision in order to overcome the problem that occurs in frames, where the flow is proportional to the angle of the obstacle - Requires supervised learning	[6, 14-16, 18-21]
Proposed approach	- A combination of laser pointers and pattern recognition module. - Training set of data of pixel intensities and corresponding distances.	-Less computation required. - Does not impose particular kind of environment. - Able to handle fast changes in shapes and overlapping colors. - Works perfectly indoors. Also works outdoors if there is no saturation in the image.	

Despite the fact that the robot is performing a simple navigation task, the control program can take a big advantage of images captured by the camera in real time. These images are used to compute a regularized depth perception that can help in navigation. Consequently, the system can be integrated in any mobile robot regardless to what the robot is primarily designed for. As long as the robot is equipped with a wireless module, the software routine will be able to perform its calculations and suggest the best move to the robot. The curvy trajectories and path waypoints that describes the theoretical maneuver route shows that the system can detect various distances by applying a prediction formula against the 2-D pixel values being read after digitizing the image into pixel locations (sampling) and values (quantization), The path has been divided into several waypoints with high rate of precision.

The distance measurement module integrated in the system showed a great amount of accuracy in real-time. The module is based on the well-known Lagrange formula. Specific pixel depth and patterns are fed to the module and converted to real distances to the upcoming obstacles as the robot moves. The results described in the evaluation indicate a high performance of the proposed approach. The estimated waypoints show that the combined module enhances detection which directly influences decisions in maneuvering and navigating.

6. OPEN RESEARCH AND FUTURE WORK

As for the future work, since the robot is equipped with a camera and is communicating with a base station, object recognition can be used as well in order to identify the surrounding objects. Therefore, an object detection routine can be added to the module.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

One of the interesting ideas would be the integration of the proposed system in cars. Modern technology in automobiles has been improved to the point by which cars can drive themselves using a group of sensors and cameras. The system would definitely benefit from the onboard cameras and would work as an assist in the function of autopilot. All modern cars are equipped with multiple computers each with a different functionality. The onboard diagnosis system (OBD-II) is the interface that can be used to communicate with the car and its peripherals/sensors. Therefore, it is possible to connect with the car's vision system. Most vision systems in this context are composed of 3 to 4 cameras on the different sides. Artificial autonomous driving is an emerging field that has been taken seriously by car manufacturers in past couple of years. Hence, communicating with the car through OBD-II and interfacing with its vision systems is the next step.

REFERENCES

[1] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J,* vol. 2, p. 00022, 2017.

[2] F. Kamil, S. Tang, W. Khaksar, N. Zulkifli, and S. Ahmad, "A review on motion planning and obstacle avoidance approaches in dynamic environments," *Advances in Robotics & Automation,* vol. 4, 2015.

[3] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," *arXiv preprint arXiv:1706.09829,* 2017.

[4] B. Zhang, L. Li, S. Cheng, W. Zhang, S. Li, and L. Xiao, "Research on Obstacle Avoidance of Intelligent Driving Vehicles Based on Monocular Camera," in *CICTP 2019*, ed, 2019, pp. 5576-5586.

[5] R. Brockers, A. Fragoso, and L. Matthies, "Stereo vision-based obstacle avoidance for micro air vehicles using an egocylindrical image space representation," in *Micro-and Nanotechnology Sensors, Systems, and Applications VIII*, 2016, p. 98361R.

[6] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1750-1757.

[7] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 593-600.

[8] S. Lenser and M. Veloso, "Visual sonar: Fast obstacle avoidance using monocular vision," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, 2003, pp. 886-891.

[9] K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li*, et al.*, "Monocular relative depth perception with web stereo data supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 311-320.

[10] F. El Jamiy and R. Marsh, "Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality," *IET Image Processing,* vol. 13, pp. 707-712, 2019.

[11] H. Okumura, T. Sasaki, A. Hotta, and K. Shinohara, "Monocular hyper-realistic AR head-up display," *Journal of the Society for Information Display,* vol. 25, pp. 34-43, 2017.

[12] P. Lima, A. Bonarini, C. Machado, F. Marchese, C. Marques, F. Ribeiro*, et al.*, "Omni-directional catadioptric vision for soccer robots," *Robotics and Autonomous Systems,* vol. 36, pp. 87-102, 2001.

[13] I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *AAAI/IAAI*, 2000, pp. 866-871.

[14] V. Grabe, H. H. Bülthoff, and P. R. Giordano, "Robust optical-flow based self-motion estimation for a quadrotor UAV," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2153-2159.

[15] C. S. Dima, N. Vandapel, and M. Hebert, "Classifier fusion for outdoor obstacle detection," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, 2004, pp. 665-671.

[16] C. S. Dima, N. Vandapel, and M. Hebert, "Sensor and classifier fusion for outdoor

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

obstacle detection: an application of data fusion to autonomous off-road navigation," in *32nd Applied Imagery Pattern Recognition Workshop, 2003. Proceedings.*, 2003, pp. 255-262.

[17] A. Vatavu, R. Danescu, and S. Nedevschi, "Stereovision-based multiple object tracking in traffic scenarios using free-form obstacle delimiters and particle filters," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, pp. 498-511, 2014.

[18] H. Mousazadeh, H. Jafarbiglu, H. Abdolmaleki, E. Omrani, F. Monhaseri, M.-r. Abdollahzadeh*, et al.*, "Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion," *Ocean Engineering,* vol. 159, pp. 56-65, 2018.

[19] A. Hussein, P. Marín-Plaza, D. Martín, A. de la Escalera, and J. M. Armingol, "Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 104-109.

[20] G. P. Stein, A. D. Ferencz, and O. Avni, "Estimating distance to an object using a sequence of images recorded by a monocular camera," ed: Google Patents, 2015.

[21] C.-H. Lin and K.-T. Song, "Robust ground plane region detection using multiple visual cues for obstacle avoidance of a mobile robot," *Robotica,* vol. 33, pp. 436-450, 2015.

[22] R. Brockers, A. Fragoso, B. Rothrock, C. Lee, and L. Matthies, "Vision-based obstacle avoidance for micro air vehicles using an egocylindrical depth map," in *International Symposium on Experimental Robotics*, 2016, pp. 505-514.

[23] M. Nieuwenhuisen, J. Quenzel, M. Beul, D. Droeschel, S. Houben, and S. Behnke, "ChimneySpector: Autonomous MAV-based indoor chimney inspection employing 3D laser localization and textured surface reconstruction," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 278-285.

[24] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 21-28.

[25] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*, 2011, pp. 2320-2327.

[26] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "J-MOD 2: joint monocular obstacle detection and depth estimation," *IEEE Robotics and Automation Letters,* vol. 3, pp. 1490-1497, 2018.

[27] H. Madokoro, K. Sato, and N. Shimoi, "Vision-Based Indoor Scene Recognition from Time-Series Aerial Images Obtained Using a MAV Mounted Monocular Camera," *Drones,* vol. 3, p. 22, 2019.

[28] *OmniVision | A leading developer of advanced digital imaging solutions*. Available: www.ovt.com

[29] *LabVIEW* Available: https://www.ni.com/en-lb/shop/labview.html