IJCTA, 10(9), 2017, pp. 693-699 ISSN: 0974-5572

The Obstacle Detection and Avoidance Algorithm for Robotic System using 2D-LiDAR and Embedded Linux

Deepali Ghorpade* and Anuradha D. Thakare**

ABSTRACT

The advancement of research on ground mobile robot as an embedded system requires high degree of automacy which is challenging in an unstructured environment, the implementation of which requires both software and hardware. The software part of embedded system consists of specific instruction sets built in as firmware. Linux which itself is a kernel, due to its acceptance and open source is developing as an architecturally neutral operating system, so Embedded Linux which denotes complete Linux distribution aimed at embedded devices is mostly used for autonomous systems. The Obstacle avoidance strategy is most significant in domain of ground mobile robot. In mobile robotic systems or autonomous vehicle's, Laser sensors (LiDARs) are most widely used for Obstacle Detection and Avoidance and mapping of environment to navigate safely through environment using rotating beams. The paper first gives overview of LiDAR sensor, its principle of operation and how they can be used for obstacle detection, navigation and path planning. The paper then proposes a design for ground mobile robotic system based on Embedded Linux platform, cross compilation and computer vision. The work mainly aims at improving performance of local Obstacle detection and Obstacle Avoidance Algorithms using 2D-LiDAR and Single Board Computer(SBC). The segmentation and clustering of the laser-point cloud data method is employed to get information of Obstacle. By using the principle of minimum cost function this method generates the forward angle and velocity of robot. The most commonly used Visibility Graph method is used for Obstacle avoidance which has advantages of good real-time performance and simple mathematical model.

Keywords: 2D LiDAR; embedded systems; obstacle detection; clustering; obstacle avoidance; visibility graph; single board computer

1. INTRODUCTION

1.1. Embedded systems

Embedded Systems [1] incorporated with Single Board Computers (SBCs) [2] are those whose functional capabilities are limited and specific. All the components of computer like the memory unit,microprocessor, input and output unit etc. are hosted on a single board in contrast to the Laptop computers and Desktop and which are fundamentally general purpose, their functionality is subject to restraints, and is embedded as a part of the complete device together with the hardware. The software part of embedded systems consist of specific instruction sets built in as firmware. Among various embedded system operating systems Linux has exploded into computing scene.

1.2. Robotic system

It integrates the electrical, mechanical, and computer science engineering technologies that deals with design, construction, operation and application of robots as well as computer systems for their control, sensory feedback and information processing. Basic Components of Robot [3] are Locomotion system,

* Dept. of Computer Engineering, Pimpri Chinchwad Collage of Engineering, Nigdi, Pune, *Email: ghdeepali77@gmail.com*

* Dept. of Computer Engineering, Pimpri Chinchwad Collage of Engineering, Nigdi, Pune, *Email: adthakare@yahoo.com*

Power supply system, Actuator system, Sensor system, Signal processing system and Control system. Robotic Vision deals with image processing and computer vision. Embedded system which consist of Raspberry pi SBC with LiDAR Obstacle detection sensor is used as Robotic system in this paper.

1.3. Navigation in robotic system

The navigation requirements can be measured by the accuracy to which the mobile robot needs to navigate. For categorizing these requirements, three terms are used:

1. Global navigation: which focuses on to determining one's position in absolute or map-referenced terms, and to move to a desired destination point.
2. Local navigation: which focuses on to determining one's position relative to objects (stationary or moving) in the environment, and to interact with them correctly.
3. Personal navigation: which Includes awareness of the positioning of the various parts that make up oneself, in relation to each other and in handling objects.

1.4. Types of obstacles

Basically, the Obstacles are divided [4] as Positive Obstacle which lie above the ground, with a positive height. And Negative Obstacles which are lie below the ground, with a negative height.

Obstacle avoidance technologies:

Broadly they are classified as

1. Local obstacle avoidance based on sensors: The most commonly used method for Local Obstacle Avoidance are Artificial Potential Field method (APF) and Visibility Graph method [5]. Both the methods have benefits of a simple mathematical equations and excellent real-time performance.
2. Global obstacle avoidance: utilizing known environment information.

1.5. Cross compilation

There are many scenarios when it is not always possible to write and build an application on the same platform. For many embedded environments, for example, the restriction on memory size both for RAM and for storage, no support for programming tools etc. Also, reasonable C compiler, the C Library and associated tools required won't fit into such a small space and developing in such an environment is obviously more difficult. So, Cross-compilers enable us to develop on one platform (the host) while building for an alternative system (the target). The target computer doesn't need to be available. All we need is a compiler that knows how to write machine code for the target platform. Cross-compilers can also be handy when there is a slow machine and requires much faster one and want to build in minutes rather than hours or days. Several different components work together, with the goal being to produce the bytecode that the CPU uses without knowing source code. [6]

The goal of this paper is to present a detailed description of the Obstacle detection and avoidance framework for Robotic System using cross compilation platform. The Raspberry pi 3 Model B (SBC) is used to implement the control logic with ARM Cottex-A7 Processor and Raspbian Operating System, the Pentium V Processor desktop computer with Linux LTS 14.04 Operating System is used as developing platform. The project will be developed in Eclipse IDE tool using C++ language on Linux Operating system 64-bit. The paper uses relatively simple and more practical Obstacle Detection method based on 2D LiDAR and Obstacle Avoidance method is modeled on results of Obstacle Detection which skillfully uses Visibility graph path planning method. The performance of Obstacle Detection and Avoidance is improved.

2. RELATED WORK

2.1. Review Stage

The goal when a robot is built is to be optimized and to be acquiescent with all specifications. To meet the requirements, we need to identify the sensors that would be the best for an application like detecting and tracking an object. Among all sensor types that can be used for target detection and tracking [4] like light sensor, image sensor, ultrasonic, infrared, sonar laser the Laser sensor is very precise in measurement so is best choice for tracking and detection a target located at a long distance. Almost all robots today use lasers for remote sensing. LiDAR (Light Detection and Ranging) is a remote sensing technology which measures distance by illuminating a target with laser and analyzing the reflected light.

Working principle: Internally LiDAR is composed of range measurement sensor that repeatedly transmits pulse of light. This pulse of light hits target then bounces off and return to range measurement sensor .BY measuring how long it takes for light to travel out and return, the sensor can determine distance to object. Additionally, range measurement sensor is mounted on spinning platform that allows device to take range measurements at many points around a 360-degree sweep. As range measurement sensors rotated, range readings are taken rapidly (up to 1000 samples per sec), this gives 2D or 3D view (depending upon type of LiDAR used) of entire surrounding of robot. [5]

The paper [4] focus on detection of Negative obstacles using Novel Setup Method of 3D LiDAR in field environment. The Geometrical Character approach is used for Obstacle Detection. Width and back of negative obstacle are considered as geometrical characters. Strength are shrinkage in blind region around vehicle and improvement in scan lines density. The contribution of work in paper [9] is to increase resolution of 3D range images by assessing unknown points in environment. Papillious Gerg algorithm is used for super resolution and inpainting, extended P-G algorithm is used for smooth obstacle boundaries capturing and Probabilistic Obstacle Reconstruction algorithm is used for very quick prediction about existence and boundaries. Strengths are to eliminate ripples occurring at edges of obstacle to obtain smooth obstacle boundaries and very fast prediction about boundaries of obstacles by using sparse scan data. The paper [5] has proposed relatively simple and most practical local obstacle detection and avoidance methods based on 2D LiDAR. The Filtering and Clustering algorithm is used obstacle detection; Visibility graph path planning method is used for Obstacle avoidance. Strengths are simple mathematical model, good real time performance. The survey paper [10], mainly discussed and analyze different algorithms for robot navigation with obstacle avoidance. The comparison of all algorithms characteristics, advantages and disadvantages is made and concluded that “Follow the gap” method is improved algorithm over others since it takes short time to reach the destination point and without requirement of any dedicated software or extra memory but has a limitation (due to its local characteristics) that it lack behind to avoid U and H-shaped obstacles. So, there is a need of an algorithm which cannot get trapped into local minima and can able to tackle obstacles of U and H shaped.

3. PROPOSED DESIGN

The Obstacle detection and avoidance algorithms used for navigation of Robotic System are improved to accurately locate obstacle and avoid them to reach to destination point using shortest path method.

The project will be developed in Eclipse IDE tool using C++ language on Linux Operating system 64-bit. Official Raspbian (armf)cross compiling tool chain is used to port binary on Single board computer (Raspberry Pi-3 Model B) which is “gcc-linaro-arm-linux-gnueabihf-raspbian-x64”. The applications of system are Underground mine navigation or mapping using robotics, in Military applications, in Industry assembly line, in Automotive production.

3.1. Architecture of system

Input and outputs:

Input given to system will be source and destination point for navigation. Output of designed system will be angle of orientation given to actuator(motor)to navigate robot.

1. LiDAR: It is used to accumulate data from the environment. Output of LiDAR will be cost map or point cloud data representing complete map of surroundings of Robot whichwill provide necessary information for system for obstacle detection.
2. Obstacle detection Strategy: The output from LiDAR includes unnecessary information for the obstacle detection so appropriate Image processing techniques are applied to extract data to apply Obstacle detection algorithm which will give exact position of Robot with respect to obstacles and detailed surrounding map around Robot.
3. Obstacle avoidance: This algorithm requires knowledge about the actual position and the area, the robot in which it is operating, which is given by obstacle detection algorithm. Path planning methods are required fix the route of Robot, Visibility graph approach is used to decide shortest route to be followed to reach to destination point. The algorithm set robots current travelling direction so that Robot will not collide with Obstacle.
4. Finally depending upon angle of orientation the Actuators are directed to achieve Robot navigation to reach destination.

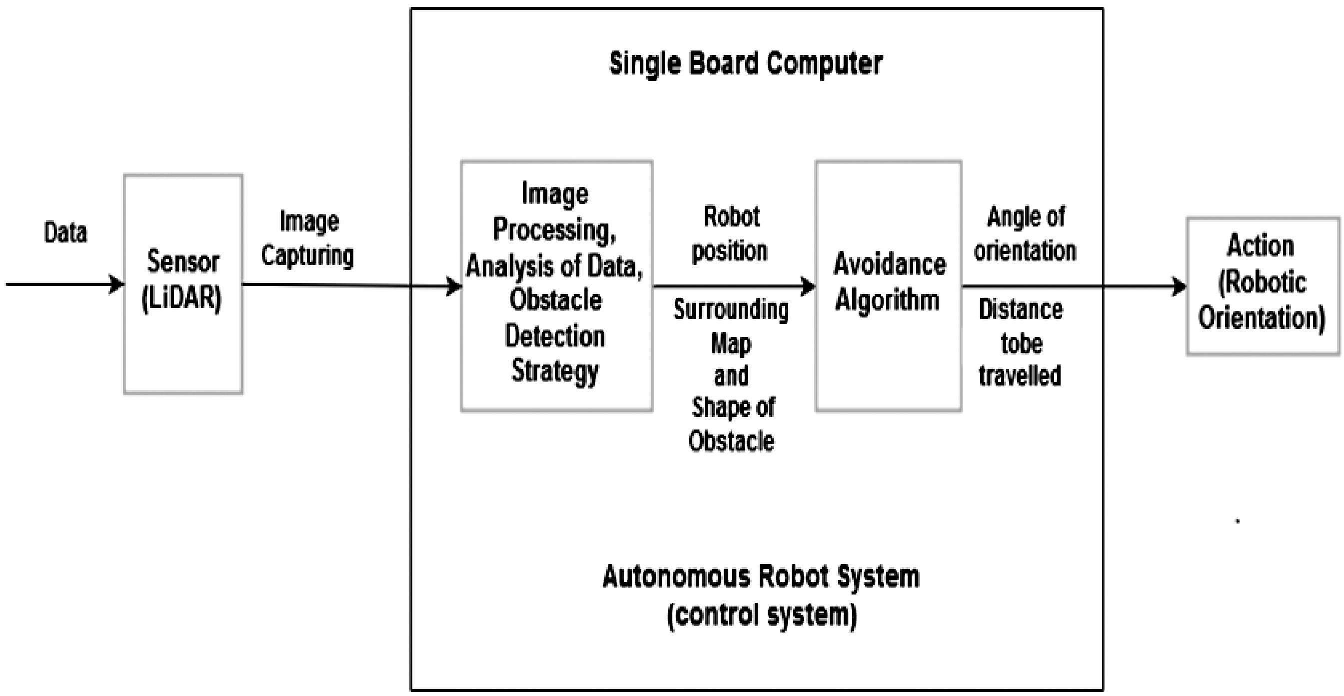

Figure 1: Block diagram of system

3.1.1. The Visibility Graph method for Path Planning

The paths are defined by the geometry of the obstacles. In the case of a Visibility graph method the paths are minimal length solutions and the edges of an obstacle result in the path itself. The aim of a path planning is to find the shortest way from starting point X to target point Y bypassing the obstacles. This is done by connecting each edge point of the obstacles with every visible edge point of the other obstacles and the points X and Y.Including the edges of the obstacles several possible paths from X to Y are developed and therefore, after establishing the paths it is necessary to select the shortest one which is done by using the Dijkstra’s algorithm which weights the distances between each edge point, checks several combinations and finds the shortest path and tries to minimize the travelling distance.

Using this approach, it is necessary to increase the size of the obstacles. Otherwise the robot would hit the obstacles.

3.1.2. Obstacle Detection Algorithm

Point cloud output from LiDAR sensor provide necessary data for Obstacle detection algorithm. The algorithm first filters the data then splits laser-point cloud data and clusters laser-points into linear, rectangular and circular Obstacles. The outcome of method gives position and shape of obstacle.

Algorithm steps:

1. Filtering: Medium filter is used to filter out noise from laser-pointdata which reducing the complexity of obstacle clustering.
2. Preprocessing: Laser-point cloud data is divided into independent segments by grouping of 381 laser-points into one block, thus forming N blocks.
3. Clustering: Each block is clustered into 3 types: Liner, Circular, Rectangular to predict shape of obstacle from which path planning method can get edges of obstacle to from path. The number of points in block and distance between two points in block is considered to cluster them to and detect shape of obstacle from which angle of orientation can be set.

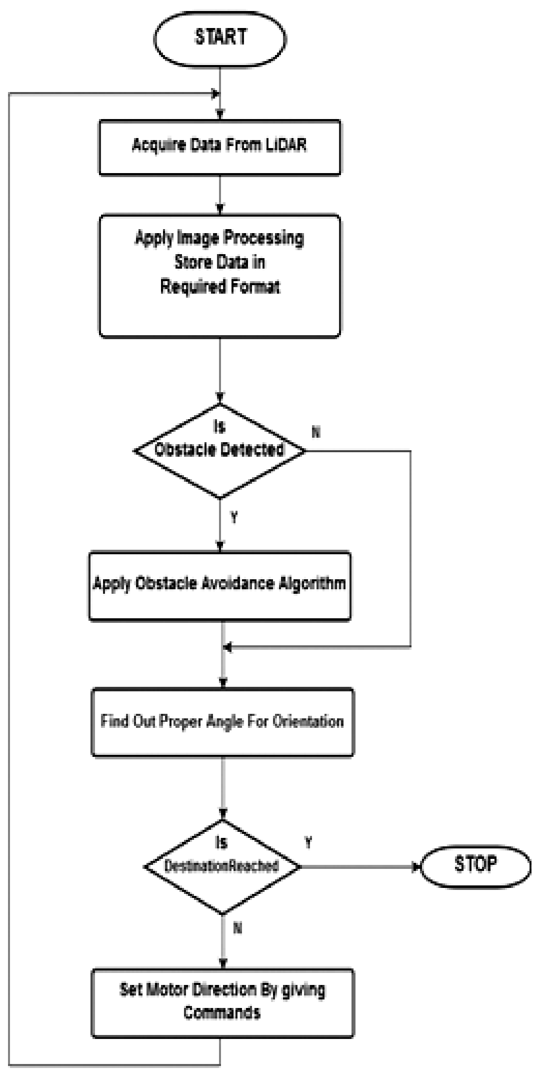

Figure 2: Flow chart of system

3.1.3. Obstacle Avoidance Algorithm

steps:

1. Set robots current travelling direction angle as (β) with no obstacle in path anddestination point coordinates.
2. Set optimum direction without any obstacle as (α).
3. Then next steering angle of robot will be β-α.
4. If robot encounter obstacle in direction áthen search steering angle as(⊖) inboth direction of α which willhelp to avoid obstacle.

4. CONCLUSION

Thus, the paper proposes a design for ground mobile robotic system based on embedded Linux platform for local obstacle detection using 2D LiDAR. The Raspberry pi 3 is used as embedded system to implement logic thus making system more robust and portable. Cross compilation is used to improve processing speed of algorithms used. The Obstacle Detection method using 2D LiDAR can filter and segment laser-point cloud data quickly and effectively and cluster the laser-points into linear, rectangular and circular shapes. The Paper also improves the Obstacle Avoidance method based on visibility graph method in complex scenarios. Future work will focus on implementation of proposed system using cross compilation platform and evaluate its performance. GPU (Graphical processing unit) can be used to achieve Parallel processing to increase processing speed drastically.

REFERENCES

[1] Jerome Leboeuf-Pasquier, Arturo Gonzalez Villa, Kevin Herrera Burgos, Donald Carr-Finch, "Implementation of an embedded system on a TS7800 board for robot control" 2014 IEEE.

[2] Xie Limei,Chen Ying, Kang Rui, "Failure Mode, Mechanism and Effect Analysis for Single Board Computers", IEEE 2011 Prognostics & System Health Management Conference.

[3] http://maxembedded.com/.

[4] Erke Shang, Xiangjing An, Member, Jian Li and Hangen, "A Novel Setup Method of 3D LIDAR for Negative Obstacle Detection in Field Environment".

[5] Yan Peng, Dong Qu , Yuxuan Zhong, Shaorong Xie, Jun "The Obstacle Detection and Obstacle Avoidance Algorithm Based on 2-D Lidar" Proceeding of the 2015 IEEE International Conference on Information and Automation Lijiang, China, August 2015.

[6] ibm.com/developerWorks, "Build a GCC-based cross compiler for Linux" Tutorial.

[7] https://en.wikibooks.org/wiki/Robotics/Sensors/Ranging_Sensors

[8] Max Meusburger, Dominik Hoerburger, Walter Goritschnig, Peter Goritschnig, "LIDAR Sensors Advanced Sensor Usage in Botball - A Concept Botball 2016."

[9] Bengisu Ozbay y , Elvan Kuzucu y, Mustafa Guly, Dilan O zturky, Muhittin Tascýy, A. Mansur Arýsoy, "A High Frequency 3D LiDAR with Enhanced Measurement Density via Papoulis–Gerchberg" 2015 IEEE.

[10] M. Zohaib1, M. Pasha1, R. A. Riaz1, N. Javaid1, M. Ilahi1, R. . Khan2 "Control Strategies for Mobile Robot with Obstacle Avoidance

[11] Robotics/Navigation - Wikibooks, open books for an open world

[12] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a onward- looking LIDAR sensor," IEEE Transactions on Vehicular Technology, vol. 61, no. 3, pp. 971-985, March 2012.

[13] M. Darms, P. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 3, pp. 475-485, Sept 2009.

[14] Halil Onur S¸ irinz and Ismail Uyanıky "Measurement Density via Papoulis–Gerchberg"

[15] Shirong Liu, Limbo Mau, Jinshou Yu, "Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems", Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference.

[16] Liu Daxue, Song Jinze, Chen tongtong, "3D LIDAR-based ground segmentation with l1 regularization", TCCT, CAA Control Conference (CCC), 2014 33rd Chinese.

[17] Pierre Luc Richard, Nicolus Pouliot,serge Mountambout, "Introduction of a LIDAR-based obstacle detection system on the LineScout power line robot"Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference.

[18] O. Yalcin,As. Sayar,O.F. Arar, S Apinar, "Detection of road boundaries and obstacles using LIDAR", IEEE Computer Science and Electronic Engineering Conference (CEEC), 2014 6th.

[19] Pavel Andris, Karol Dobrovodsk´, "Developing an Embedded System Based on a Real-Time Version of Linux" 2014 IEEE 23rd International Conference on Robotics in Alpe-Adria-Danube Region.