Simultaneous Localization and Mapping for Automatic Chair Re-arrangement System

Prof. Swapna Borde[1], Reuben A. Fernandes[2], Manoj S. Goregaonkar[3], Kapilesh D. Gujarathi[4]

[1,2,3,4] *Department of Computer Engineering, VIDYAVARDHINI'S COLLEGE OF ENGINEERING AND TECHNOLOGY K.T. MARG, VASAI ROAD (W), DIST-PALGHAR, PIN: 401202.*

---***---

Abstract- *Mapping is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. Localization means the process of determining the location of an agent with respect to some know reference .These techniques are cumulatively termed as SLAM (simultaneous localization and mapping), and can be used for the automation of any system. SLAM is a topic of ongoing research in computer technology. There are many know techniques to implement SLAM. Our goal is to demonstrate a SLAM technique by implementing it in an "Automatic Chair Re-arrangement system". This system automates the movement of a chair so that it can move back to its original location post being displaced.* [4]

Keywords- Mapping, Localization, Automation, SLAM, Chair.

1. INTRODUCTION

THE solution to the simultaneous localization and map building (SLAM) problem is, in many respects, a "Holy Grail" of the autonomous vehicle research community. The ability to place an autonomous vehicle at an unknown location in an unknown environment and then have it build a map, using only relative observations of the environment, and then to use this map simultaneously to navigate would indeed make such a robot "autonomous". Thus the main advantage of SLAM is that it eliminates the need for artificial infrastructures or a priori topological knowledge of the environment. A solution to the SLAM problem would be of inestimable value in a range of applications where absolute position or precise map information is unobtainable, including, amongst others, autonomous planetary exploration, subsea autonomous vehicles, autonomous air-borne vehicles The general SLAM problem has been the subject of substantial research since the inception of a robotics research community and indeed before this in areas such as manned vehicle navigation systems and geophysical surveying. A number of approaches have been proposed to address both the SLAM problem and also more simplified navigation problems.

2. EXISTING METHODOLOGIES

2.1 Dead Reckoning:

Dead reckoning, a method of navigation that uses on-board sensors to measure speed and heading based on the last known position to estimate its current position, becomes important. On-board sensors, gyroscopes and a compass, provide vehicle attitude and orientation information. Speed is either measured in real time or is determined empirically through testing. When accelerometers are available on-board, a more advanced form of dead reckoning known as inertial navigation is possible. Inertial navigation integrates accelerations and rotation rates of a body in time to determine attitude and heading. [1]

Drawbacks:

Errors in dead reckoning result from many sources, including inaccurate knowledge of the vehicle's starting position, a misalignment of the heading sensor, and the time-varying effects of the sensors. Any errors present in the initial conditions will be propagated throughout the vehicle run. Similarly, a misaligned compass causes a constant offset in all measurements and translates over time into an estimated heading trajectory that deviates from the actual vehicle trajectory. Finally, the constant drift in gyroscopes becomes significant with time.

2.2. Odometry:

Odometry is the use of data from motion sensors to estimate change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location. Each motor has a rotary encoder, and so one can determine if either wheel has travelled one "unit" forward or reverse along the floor. This unit is the ratio of the circumference of the wheel to the resolution of the encoder. [2]

Drawbacks:

The agent will always trace back the path described by it to reach its current location. However if the agent were displaced in an indefinite path, instead of following a straight path to its original destination it will retrace that path again, which is a waste of time and energy. Also if the agent were displaced by not dragging but by lifting the chair, then the rotary motor will not read any change in location and hence will not able to retrace its path to the destination.

2.3. Position triangulation using BLE beacons:

BLE Beacons are small devices available in a wide range of shapes to be mounted on walls, tables, etc. These devices are specially designed for indoor locations. A robot can detect the BLE beacon signal and calculate its position in the range of more than two beacons and estimate the location. The beacons can run on a single battery charge for years, and this is one of its advantages in front of other localization systems. Using BLE beacons to calculate the indoor position should be easier, at least in theory. The robot receives tiny and static pieces of data within short distances. [3]

Drawbacks:

The accuracy of the pose estimation is only as good as the camera being used. The tag detections were not able to give reliable detections of tags, and when the camera was being moved, the results worsen. Motion affects the accuracy of the pose estimation. When the robot is moving, especially while turning, it was found that the pose estimation could be off by > 20 degrees, and vary in range by up to 0.75 meters.

2.4. April Tags:

April Tags is a visual fiducial system, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration. The tags provide a means of identification and 3D positioning, even in low visibility conditions. The tags act like barcodes, storing a small amount of information (tag ID), while also enabling simple and accurate 6D (x, y, z, roll, pitch, yaw) pose estimation of the tag. [5]

Drawbacks:

This is a relatively good method but it was ruled out since the selected method is easier to implement and is less costly.

2.5. Global Positioning System (GPS):

The GPS system currently has 31 active satellites in orbits inclined 55 degrees to the equator. The satellites orbit about 20,000km from the earth's surface and make two orbits per day. The orbits are designed so that there are always 6 satellites in view, from most places on the earth.The GPS receiver gets a signal from each GPS satellite. The satellites transmit the exact time the signals are sent. By subtracting the time the signal was transmitted from the time it was received, the GPS can tell how far it is from each satellite. The GPS receiver also knows the exact position in the sky of the satellites, at the moment they sent their signals. So given the travel time of the GPS signals from three satellites and their exact position in the sky, the GPS receiver can determine your position in three dimensions - east, north and altitude.[6]

Drawbacks:

GPS' accuracy is not precise enough to determine the location of the agent.

3. IMPLEMENTATION METHODOLOGY

We use an overhead camera that can view the entire area on which the agent will move. The camera (using OpenCV) can recognize the agent as a solid color (any dark color other than red).The computer uses a code (Python 2.7 and OpenCV code) is used to determine the angle between the target (original location represented by a "Red dot").The agent has a magnetometer which gives the heading of the bot w.r.t north (Fig I) (which is set to top of the computer screen and not geometric north). This heading is given to the computer program. The program is coded to spin the bot so that the angle inscribed with north is equal to the angle made by the bot and the target. At this point, move the bot towards the target. When the bot overlaps the target, the movement is stopped. Using proximity sensors and IR sensors, the bot a rotated about its position until it is properly oriented. [7]

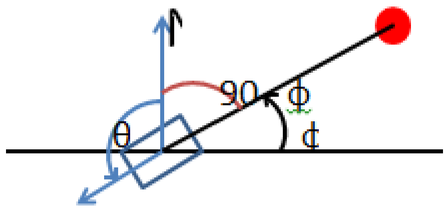

Fig 1: The red dot shows the target location to arrive at. Blue dot represents the robot.

3. DESIGN OF THE SYSTEM

3.1. CIRCUIT DIAGRAM

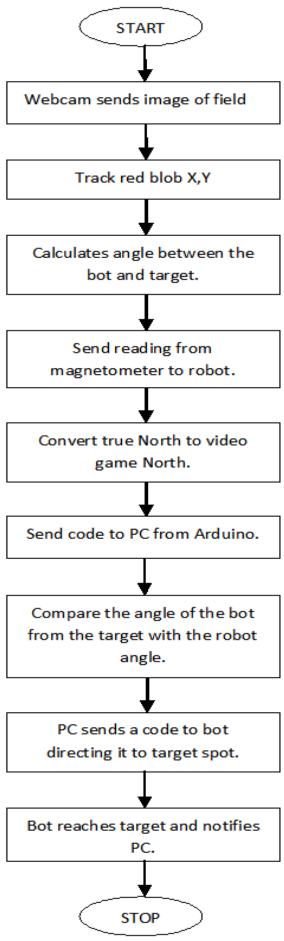

Fig 2: The flowchart clearly shows a step by step functioning of the entire system.

Webcam sends images of its field-of-view. OpenCV looks for the largest red blob. It begins tracking the red blob's X, Y. The PC averages these X, Y positions for around 150 camera frames. If the blob hasn't moved much, the PC assumes the red blob is the robot. The PC gets frisky and gives our robot a random target within the webcam's field-of-view. The PC calculates the angle between the bot and the target. Meanwhile, the robot's microcontroller is taking readings from a magnetometer on the robot. The robot, with a one time human calibration, translates true North to "video-game north," aka, top of PC's screen. The microcontroller transmits this code to the PC. The PC compares the angle of the bot from the target with the robots angle. The PC sends a code to the bot telling it to turn left, right, or move forward (closer to the target).When the robot has made it within an acceptable distance from the target he "Munches the Dot." A new random dot appears. Rinse repeat.

3.2. BLOCK DIAGRAM:

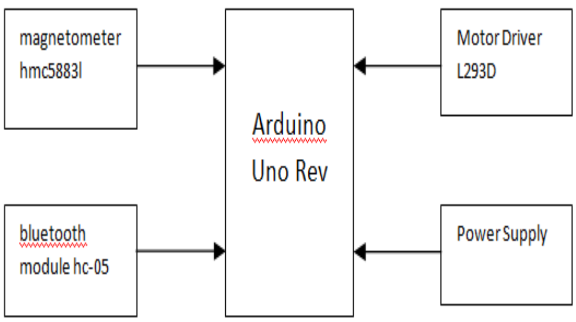

Fig 3: Figure illustrates the major blocks of operating hardware components.

The arduino is the main processor, magnetometer is used to find heading of the robot, HC-05 Bluetooth module for communication with the computer and L93D motor shield for voltage regulation of power supply

4. CONCLUSION

The objective of this project was to develop a technology by which smart AI driven systems can communicate and access the real world environment. This has been achieved by implementing SLAM, by using image processing. This system can control multiple robots as long as they are in its field of vision. The proposed method has been implemented for a “self-parking chair”, however, various other implementations such as automated vacuum cleaners, automatic parking system for cars and other robots in general. This system requires a closed environment in which the camera can constantly see the environment in which the target object must move.

5. REFERENCES

1. https://www.sciencedirect.com/science/article/pii/S1474667017440882
2. Daniel Pizarro ⋆, Manuel Mazo, Enrique Santiso, Marta Marron, David Jimenez, Santiago Cobreces and Cristina Losada, “Localization of Mobile Robots Using Odometry and an External Vision Sensor” www.mdpi.com/journal/sensors, 13thApril 2010.
3. Hyunwook Park, Jaewon Noh, Sunghyun Cho,” Three-dimensional positioning system using Bluetooth low-energy beacons” https://doi.org/10.1177/1550147716671720.
4. https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
5. https://openmv.io/blogs/news/apriltag-marker-tracking-the-future-is-here
6. https://en.wikipedia.org/wiki/Global_Positioning_System
7. https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/