Autonomous Mobile Robot Navigation and Localisation using Cloud, based on Landmark

[1]Dhruba Ningombam, [2]Prerna Rai, [3]Tejbanta Singh Chinghtham

Sikkim Manipal Institute of Technology

ABSTRACT— *Robotics is one of the fields which have evolved greatly during the past years. Years back robots were mostly independent and performed task on a microchip. This created a large overhead on their processing power. Therefore today the robot has shifted its computation and processing to the cloud rather than performing task independently. The study here focuses on landmark and text identification in the camera captured image using Representational state transfer (REST) application program interface (API). The robot than navigates based on the information in its environment representing a path map in simple format. The generated map is stored in the cloud environment and any other follower robot can access the map using on demand resources sharing property of cloud. The paper also provides study related to localization of follower robot. It needs to localize itself relative to the accessed map to perform navigation. For this purpose Markov localization algorithm has been used. The algorithm has the capability to deal with ambiguous situations and can recover the position of robot in case of localization failures*

KEYWORDS—Cloud;Robotics;Landmark; Navigation;

I. INTRODUCTION

The concept of robotics augmented with cloud has brought about a great functional change in today's era. Robot can be made intelligent like human beings[13]. They can be made to take decisions[13]. This decision making process heavily relies on machine learning algorithms available as a service in the cloud. Machine learning is the idea where generic algorithms can provide useful information about a set of data without having to use specific code. Instead of writing code, the generic algorithm builds its own logic based on the data [1]. Machine Learning Algorithms are of two types. They are

a. Supervised learning and

b. Unsupervised learning**.**

Supervised learning is a process in which the algorithm learns from training datasets. Some of the supervised learning problems are Classification and Regression. On the other hand unsupervised learning algorithms discover and present the required structure of data on their own. Some of the unsupervised learning problems are Clustering and Association [1]

The work that has been discussed in this paper makes use of the machine learning algorithm via Google cloud vision API. Cloud vision API provided by Google helps developers in understanding the content of an image using machine learning models in an easy-to-use REST API. The image uploaded to the cloud is classified into different categories based on the service used such as detecting landmarks or text or label in the given image[2]. Among many services of cloud the project makes use of landmark detection and text detection API. The detected outcome is used by the robot to navigate in a particular environment. When an image of popular landmark is send to the API it returns JavaScript Object Notation (JSON),with the name of the landmark, its latitude / longitude coordinates, a bounding box indicating where the landmark was found in the image, and even more landmark metadata[3]**.** In order to carry out identification of landmark, the cloud references a database of over thousands of landmarks. The landmark database includes the name, location and the id of an entity. The landmark data is updated continuously in order to include new landmarks.

When a real time image captured by robot is sent to the landmark recognition endpoint in the cloud, the clusters of image is used as a model image and the given image is matched with the model image using nearest neighbor principal [3]. A match score is than generated. If the match score is above threshold level, it is identified as landmark. The navigation of robot is not only based on detection of known landmark but also on detection and extraction of text in the given image.

For the text detection and extraction in the image, optical character recognition is performed with the support for a broad range of languages. Texts present in camera captured images are considered to be one of the most important and strong sources of information about that image and about the place or situation from where the image was captured [4]. Therefore the text extraction service can help automate robot to navigate in an environment.

The paper is focused to make any robot follow a path generated by a master robot. The master robot simultaneously generates a map for the path by performing navigation using cloud service. It detects the landmark that comes ahead of the robot using landmark detection service and text detection and extraction service of Google cloud. The map is stored in the cloud and can be accessed by any robot to follow the path taken by master robot. This is how the knowledge acquired by one robot can be shared with others. Also the other robot need to localize itself relative to the map provided. Localization means position estimation and position control. This can be achieved using Markov localization. It is based on the concept of conditional probability. Markov localization enables robot to localize even under ambiguous situation and can also recover the position of robot in case of localization failures [11].

The paper also provides an experimental process performed for the robot GoPiGo as master robot and Lego Mindstrom as the follower robot. The controlling device of the GoPiGo system is a Microcontroller, Bluetooth module, Wi-Fi module and DC motors are interfaced to the Microcontroller using GoPiGo motor control board. The use of gateway helps in communicating the cloud services to the robot. The controller acts according to the control set by the gateway and hence the DC motor of the Robot performs accordingly. In achieving the task, the microcontroller is loaded with a program written using python language in Unix platform [5]. The controlling system for Lego Mindstrom is ARM9 microcontroller with inbuilt Bluetooth module that communicates with android device such as smart phone. The smart phone acts as a middleware and allows the Lego Mindstrom to communicate with cloud and request a map. On access of the map the robot than makes necessary navigation.

II. RELATED WORKS

Many studies have been undertaken by various researchers regarding the navigation of robot using landmark. Various techniques and principles may have been used in the past. In landmark identification and navigation of robot, vision has been one of the major elements. The study in paper [6] is based on natural visual landmark using the concept of SLAM paradigm. The paper describes topographical navigation and uses Markov localization for handling ambiguous landmark. The system proposed remains effective even when there is a variation in lightning conditions but the use of SLAM do not prove effective as it could be CPU intensive and additional task is not allowed. In another paper [7] the landmark used is artificial landmark. It uses 2D bar code based on computer vision. In this approach the system creates a map for indoor environment and allows robot to navigate. The use of artificial marker makes the navigation faster and efficient but the approach does not make use of movement orientation during the robot navigation. The system developed for robot to navigate as stated in [9] have made use of kinect sensor. This sensor acquires a video and converts into 3D point cluster using clustering and filtration process. But the major drawback of the proposed system is that the quality of image captured is poor and the landmark may be misidentified due to presence of large outliers especially in large environment. In one of the other paper, as stated in [8], the navigation of robot is done using cloud. The introduction of cloud with robotics reduced the overhead caused on the robot and improved performance and computational skill [8]. The system in [8] proposes data fusion with advanced sensing system for automated guided vehicles. But the major limitation is to make the robot navigate freely by increasing the flexibility and also in classifying the object. Therefore the researcher Limosani, aims in providing navigation in complex environment in [10]. It makes use of environmental tags with cloud Software as a service (SaaS) and map supervisor for navigation of mobile robot. The AR tag and QR tag is used as environmental tags. The tag provides information about the remote resource. The system proposed makes use of artificial landmark and is not based on recognition of scene.

III. SYSTEM OVERVIEW

The system aims in implementing Raspberry Pi controller as master robot that controls the robotic movement. It autonomously navigates by identifying the landmark until the destination is reached. The navigation of robot simultaneously generates a map and is automatically stored into the cloud. The on demand cloud service facilitates the sharing of the stored map to any requesting robot. The identification and decision making is based on the application provided by Google cloud. The Raspberry Pi controller is coupled with GoPiGo motor control board that drive the motor on the decision provided from the raspberry pi using the service provided by cloud. Raspberry Pi makes an interface to the cloud so as to access and monitor the information required for robot navigation. We have made use of Wi-Fi module.

The other module is the follower robot. For the purpose of implementation Lego Mindstrom is used. The robot has an ARM9 processor with inbuilt Bluetooth module. The android smart phone has been mounted on the robot for it to act as a communication medium between cloud and the robot. The android smart phone is provided with an app that helps to communicate. It acts as a middleware between cloud interface and robot.

The system aims at developing a system to achieve the target that can provide following functionality in terms of various modules.

a. To activate gateway that provides an interface to control GoPiGo which involves forward, backward, left, right and stop navigation system.
b. GoPiGo to detect landmark by using distance measurement sensor such as ultrasonic sensor.
c. GoPiGo to capture real time image of landmark using pi camera when a landmark is detected using ultrasonic sensor.
d. To generate a map on the process of navigation.
e. To activate Lego Mindstrom that initializes the communication with cloud.
f. Download the map generated by master robot (GoPiGo).
g. Calibrate the time taken by GoPiGo to its time.
h. Follow the path generated by GoPiGo.

IV. SYSTEM ARCHITECTURE

As stated above the system has various modules. The system architecture for the navigation of robot in cloud environment is shown in the Figure 1 below. The module begins with cloud communication using gateway or middleware. The sensor is used to detect the landmark ahead of the robot which when sensed is then captured by camera and uploaded to cloud in jpeg format. And using the Google cloud services the landmark is identified and navigation proceeds creating a map. This map is stored in the cloud. This information can be shared with other robot and follow the path. The flow of control for the system is also shown in the Figure 2 below.

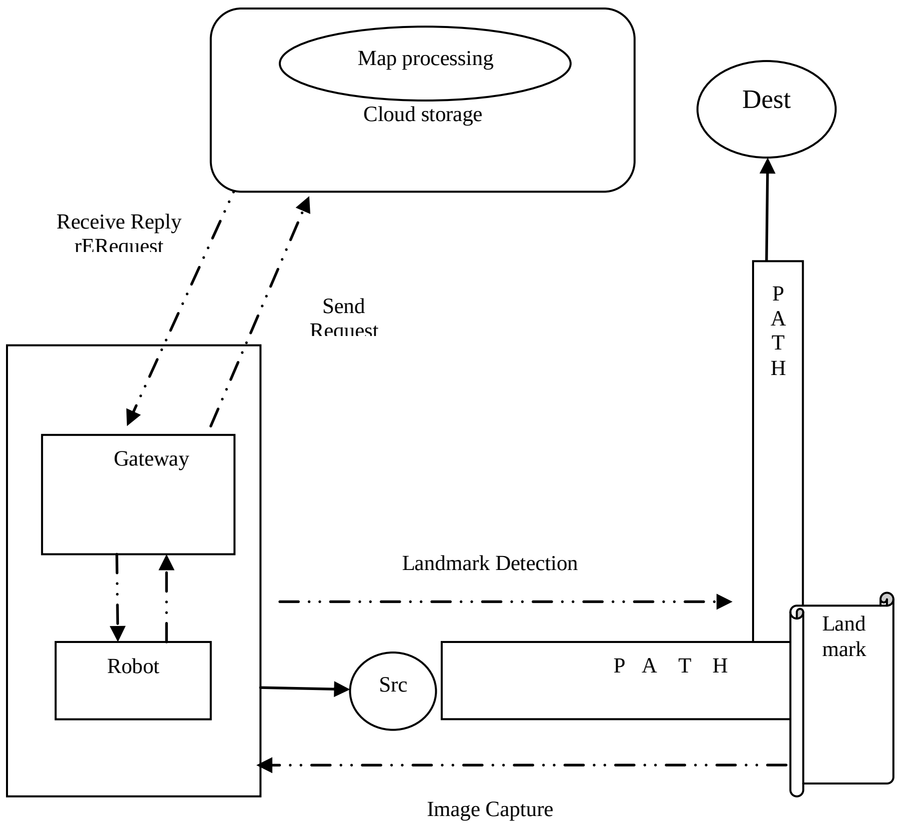

Figure 1.Cloud robot navigation using

V.IMPLEMENTATION AND SYSTEM INTERFACE

The system interface consists of master robot GoPiGo and Lego Mindstrom as follower robot. GoPiGo is provided by Dexter Industries. It consists of Raspberry Pi 3 model B, Ultrasonic sensor, Raspberry Pi camera module, Wi-Fi module and DC Motor. The robot with raspberry pi 3 acts as the base hardware which is the gateway that connects to the cloud. The cloud service that is used in the project is the Google cloud vision. The Google cloud vision provides a REST API that has an analytical capability to identify and detect natural landmark in the image captured by the raspberry pi camera. The DC motor driven by GoPiGo driver is also in synchronization with the raspberry pi which makes the robot perform the required kinds of motion. Lego Mindstrom, on the other hand initializes communication using android smart phone which acts as a middleware. It is the central mode for communication.

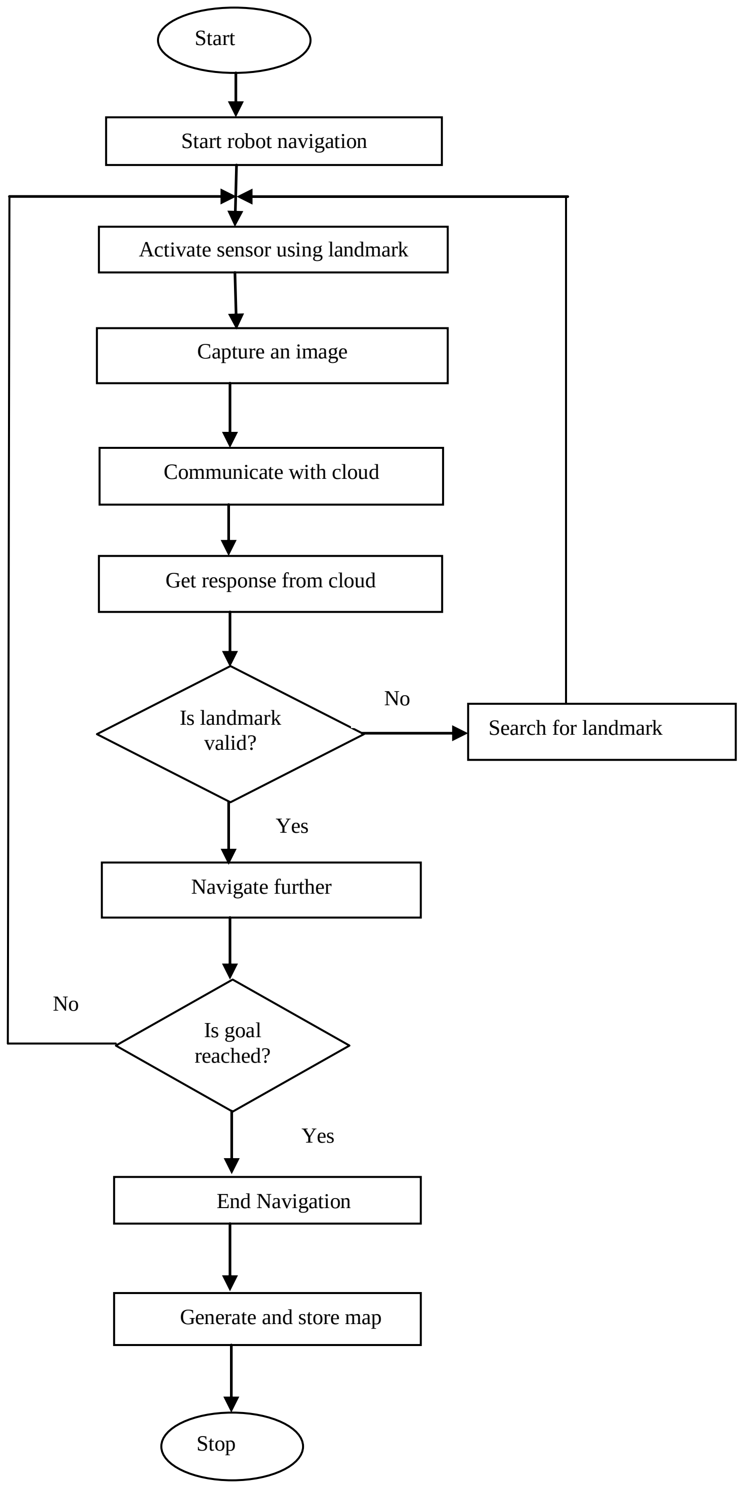

Figure 2.Flow chart for cloud robot

This communication is possible either by using Bluetooth or Wi-Fi module.

Prerequisite:

a. Wi-Fi module to be connected to raspberry Pi model B.
b. Internet connectivity to be maintained in the robot and smart phone.
c. Creating credentials for connectivity to cloud.
d. Enabling Google cloud.
e. Enabling Bluetooth in android smart phone and Lego Mindstrom.

Input:

The Image in jpg format captured from Raspberry Pi camera module acts as an input to Google cloud connected using service key credentials in the form of json file created for the specified project.

Output:

There are two distinct outputs that are generated during the process. They are

a. Cloud performs analytical capability over the image received and detects landmark and sends the output to the robot enabling GoPiGo robot to navigate further based on the decision made.
b. Simultaneously generate a path map in the form of generalized string (e.g.:10L20R10S).

The system makes use of cloud vision services that provides mechanism to understand the content of an image by encapsulating powerful machine learning model. It is an image analytical tool with pre-trained models. Using these services of cloud the image classification and detection of individual object can be done quickly. As depicted in the Figure 3 below the cloud service used is Google cloud vision. It can analyze real time images uploaded in the request provided by the GoPiGo robot controlled by Raspberry Pi. Raspberry Pi acts as a gateway between the robot and cloud. This robot is considered to be the master robot. It uses ultrasonic sensor to determine distance from a robot to the nearest landmark and also to avoid obstacles in order to navigate.

The Figure 3 shows the different states which the system undergoes. The data dictionary for the different states is as below.

1. Request for service: Google application credential creation asrobot1.json; encode and upload image.jpg,
2. Response: Description; longitude; latitude. x coordinate; y coordinate of image.jpg
3. Map upload: firstmap.txt(20L15R10S)
4. Navigate: forward( );left( );right( );stop( );backward( )
5. Request map: request firstmap.txt
6. Map download: firstmap.txt(20L15R10S)

The states 1, 2, 3and 4 are followed by master robot which is GoPiGo while the states 4, 5 and 6 is performed by Lego Mindstrom EV3 robot.

The robot GoPiGo and Lego Mindstrom EV3 is taken up for experimental purpose. In real time world the robot can be replaced by any form unless it uses a gateway or middleware.

For the experimental purpose in our system GoPiGo is used as master robot and Lego Mindstrom EV3 is used follower robot. The problem that has been defined in this system is the sharing of knowledge acquired by master robot to the follower robot. The system is also designed to demonstrate the use of different modes of communication. The communication medium used in GoPiGo is Raspberry Pi. It acts as a gateway between the cloud environment and the robotic movement control. In Lego Mindstrom EV3 the communication medium used is an android smart phone which

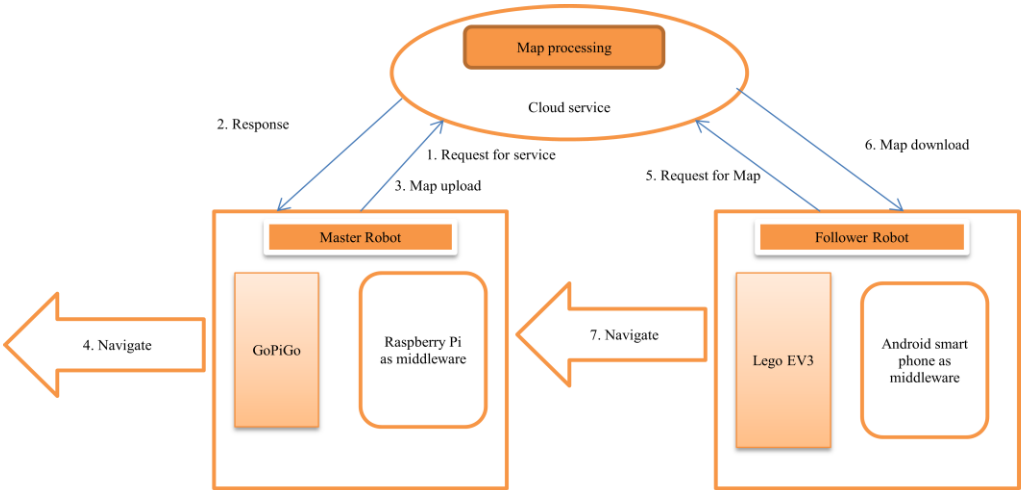

Figure 3 Block Diagram

acts as a middleware between the cloud environment and the robotic movement.

The decision for the robot to navigate in particular direction is depicted by the cloud service provider. The path determined by the master robot while navigation is recorded in the form simple string as shown in Figure 4. This string provides information about the actions needed to navigate for completing the path. The path generated by master robot is dynamically stored as per the navigational decision made by cloud. This predetermined path can now be stored in the cloud in the form of map.

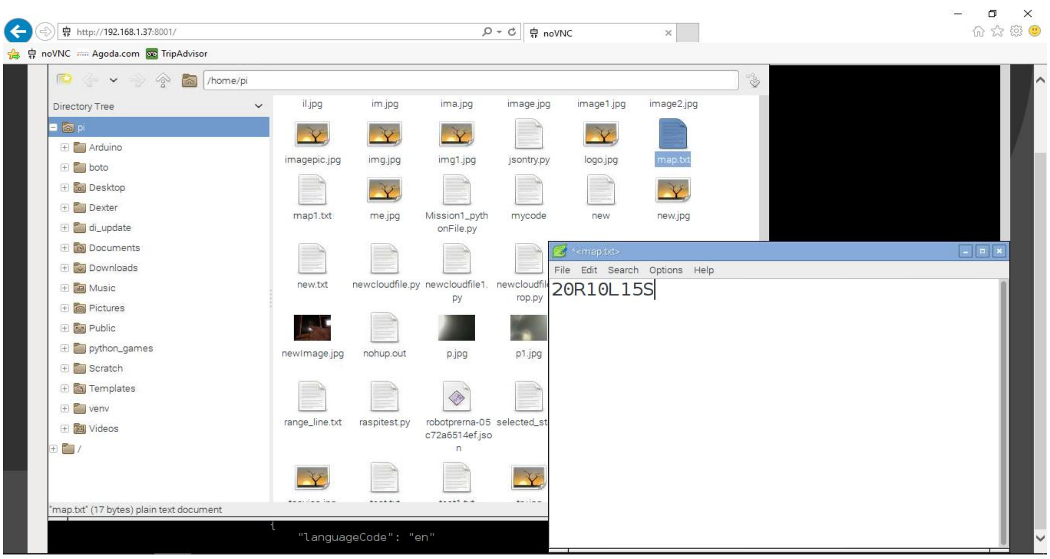

Figure 4 Generated Map.

Before the follower robot which is Lego Mindstrom EV3 receives the path that was stored it passes through an android interface. The task of the interface is to mask the heterogeneity of robots. The android device takes the representation and converts the value of the path accordingly and provides this data to the robot that needs the path.

Time T in Seconds	No of trails (Distance D in cm)				(cm)	S=D/T(cm /sec)
	Trail 1	***Trail 2***	***Trial 3***	***Trial 4***	***Average Distance D***	***Speed S***
20	200	199	200	198	199.25	9.96 (S1)
10	98	100	99	100	99.25	9.925 (S2)
5	50	48	48	50	49	9.8 (S3)
2.5	26	25	24	26	25.25	10.1 (S4)

TABLE 1. AVERAGE SPEED OF GoPiGo

VI. RESULT AND DISCUSSION

Experiment was conducted as shown in Figure 3. For the master robot GoPiGo controlled by Rasberry Pi a practical approach of determining the speed of robot was verified with the hardware configured motor and its actual movement. It also determines the range of the sensor at which the landmark will be detected and identified accurately.

A. Average speed of GoPiGo robot

The average speed of GoPiGo robot is determined by making it undergo forward movement number of times while giving various time periods for the motor movement as shown in the Table 1. The average speed of robot is determined using two methods.

Method 1:

The practical approach determines the average speed taken by GoPiGo robot by finding out the average of speed S taken for four different time periods as per the equation below:

Average speed= (S1+S2+S3+S4)/4 (1)

Therefore Average speed= 9.946 cm/sec.

Method 2:

Determining a method 2 to determine an average speed taken by GoPiGo while given the following values for the wheel

Given values:

i. Diameter (D) of wheel of robot=6cm

Circumference(C) of a circle= π. D cm (2)

Using (2)

$C = \frac{22}{7} * 6\text{cm}$

=18.85cm

ii. Rotation per minute (rpm) of wheel of GoPiGo= 30rpm

Assuming 10 sec the number of rotation is 5

Therefore 1 sec= $\frac{5}{10}$ = 0.5 rotation

Hence Average speed (S) of GoPiGo is = $\frac{5}{10} * 18.85$ cm

= 9.425 cm

It is seen from the above method 1 and method 2, the average speed of GoPiGo is approximately equals to 9.6 cm

B. **Accuracy range for the identification of landmark**

Test was performed on GoPiGo to identify the range of distance at which the ultrasonic sensor can detect the landmark and capture an image accurately. The following Table 2 and Figure 5show the accuracy in capturing an image of landmark from a range of distance and in identifying it. The study depicts that accuracy in detecting landmark is highest within the range of 41cm to 70 cm.

TABLE 2 ACCURACY RANGE

Sl. No	Ultrasonic distance range	Landmark identified out 0f 10	Accuracy in %
1	0-10	Nil	Nil
2	11-20	Nil	Nil
3	21-30	Nil	Nil
4	31-40	7	70
5	41-50	9	90
6	51-60	10	100
7	61-70	10	100
8	71-80	7	70
9	81-90	2	20

C. **Calibration of time for follower robot**

The decision making for the follower robot is based on the decision made by the master robot. The path generated by master robot is requested by the follower robot to perform the course of action accordingly. The path is in the form of simple strings which include the time taken for the wheel of robot to drive to the landmark and make a decision to navigate left(L) or to right(R) or forward (F) or stop (S) i.e 10L20R10F15S.

Since the wheel diameter of every robot may differ before the robot receives the path it passes though android interface. The task of interface is to mask this heterogeneity of robot. Following is the required calibration needed for every follower robot to perform before navigation.

Let the time taken by follower robot is T2

Assume:

i. wheel diameter = x cm

ii. Rotation per sec= y rotation/sec

Therefore:

Distance (D) in 1 rotation= $\pi * x$cm (3)

And:

Distance (D) in 1 sec=$Y * \pi * x$cm (4)

Hence:

$\frac{D}{9.425} = \frac{T2}{T1}$

$T2 = D * \frac{T1}{9.425}$ (5)

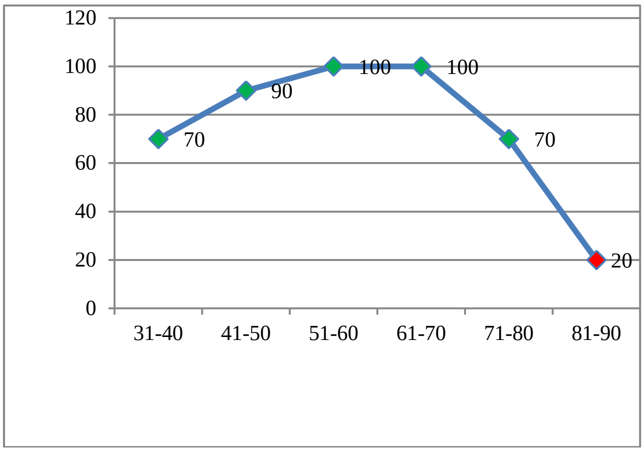

Figure 5 Graph for accuracy of landmark identification

From the above equation (5) we can deduce the time taken for any robot based on the master robot. The findings state that any robot can be made to follow the master robot by calibrating the average speed based on the diameter of the wheel and the rotation of wheel per minute.

D. **Surface area covered by Raspberry Pi camera for viewing**

The experiment taken up determines the surface area covered by Raspberry Pi. The area is in the form of trapezoid named as ABCD as shown in the Figure 6. This study provides information about obtaining the accurate landmark from the captured image increasing the performance of the system.

A mathematical calculation as per Figure 6 is performed in the following equation determining the surface area for a Raspberry Pi camera to view the image precisely and accurately.

In Δ OAF and ΔODE

From the similarity Δ

$\frac{AF}{OF} = \frac{DE}{OE}$ (6)

DE = $\frac{\boldsymbol{AF\ X\ OE}}{\boldsymbol{OF}}$ (7)

= $\frac{\mathbf{57.25\ X\ 48}}{\mathbf{100}}$ = 27.48

DC= 2 X DE (8)

= 2 X 27.48

= 54.96

In trapezoid ABCD

CD=54.96 cm

AB= 114.5 cm

EF=52

Therefore Area of trapezoid ABCD = $\frac{1}{2}EF(AB + CD)$ (9)

= 4405.96 cm^2

Therefore static area covered by Raspberry Pi camera of GoPiGo robot at particular instance is 4405.96 cm^2

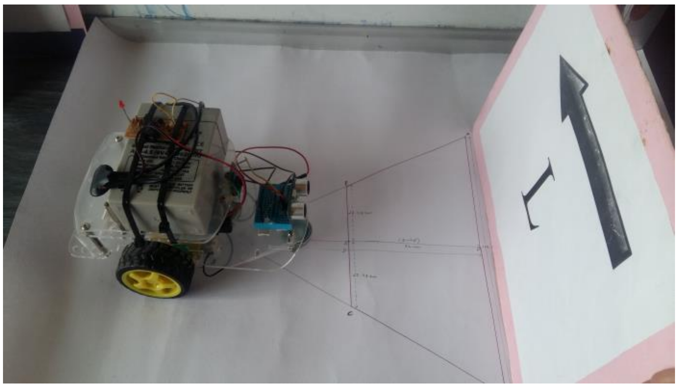

Figure 6 .View of image by raspberry pi camera

VII PATH REPRESENTATION AND LOCALIZATION

The robots used for the experiment purpose is not homogeneous in nature so the architecture and the support for languages varies to a great extent.

Therefore a uniform path representation is followed. The path is represented in the form of 20L30R10F10B15S where

L=left

B=backwards

R=right

F=forward

S=stop

The above mentioned string will cause the robot to move 20 sec left, 30 sec right, 10 sec forward, 10 sec backward and then stop. Representing in strings makes it possible to store the entire path in the representation.

As the robot finds the paths and store them in the cloud, eventually there will be huge collection of paths forming a big data. These huge data sets cannot be processed using traditional computing techniques. The proposed system does not provide an approach to handle big data sets. In future the same system can be extended further in solving the problem by using parallel processing. The system can make use of Hadoop Map reduce. Map reduce is a programming model that can be used for processing large data sets. Programs written using this model are highly parallelized. The task of partitioning the input data, scheduling the programs execution in different clusters, making it fault tolerant and managing inter machine communication [2].

For using mobile robot successfully, effective navigation is a basic and initial condition. The robot must localize itself to represent the environment in terms of map and to carry out navigation plans. Therefore localization is one of the major areas of concern in robotics. The approach used in this project for mobile robot localization is Markov localization which addresses all aspects of localization.Markov localization allows for localizing starting from any unknown position and can recover from ambiguous situations because the robot can track multiple, completely disparate possible positions.

As stated in [11], Markov localization algorithm is an efficient algorithm for the estimation of robot position as it senses and moves in the particular environment. The algorithm is based on the concept of conditional probability within the environment based on various interpretations. According to the study in [11] a probability distributions representing the robots belief of being at location l is given as $BEL(L_t=l)$.

The belief state of robot to be in location l stated as $BEL(L_t=l)$ is based on two models. They are action model (a_i) and percept or sensor model (s_i). a_i and s_i are the values that is observed. The observed value for sensor model can either be distance measured by sensor or abstract features such as room or corridor or doorway. While the observed values for action

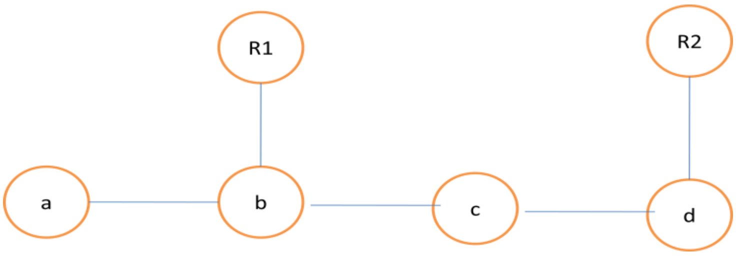

Figure 7 Topological Map with nodes.

model can be the values measured by the robots wheel encoders.

In our application Markov localization has been implemented using topological representation of belief state where each possible location l corresponds to the node in a topological map of the environment as shown in Figure 7. Due to the nature of this representation the percepts are abstract features extracted from Raspberry Pi camera. In accordance to [11] Dervish represented the abstraction of the real world based on a quantization of the world into a set of states; each state corresponds to a node on the topological map or a passage segment between two nodes.

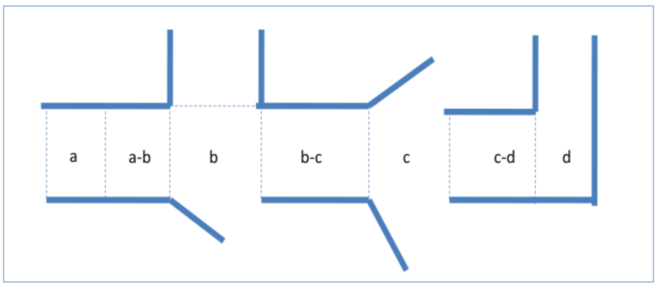

Figure 8. Quantization of passage

In our project the concept of Dervish as in [12] has been applied as shown in Figure 8. It depicts the quantization of a section of passage. The nodes in the passage are labelled with a, b, c, d whereas the area between two nodes is labelled with a dashed character. The state set represents positional uncertainty because it captures all the actual positions at which the robot believes it could be located. Each state is associated with *certainty factor*, computed using *certainty matrix* given in Table 3. The certainty matrix represents the likelihood of the robot being in the subsequent state.

TABLE 3 CERTAINTY MATRIX

	Wall	Closed room	Open room	Open passage	Lobby
Nothing detected	0.80	0.40	0.04	0.005	0.20
Closed room detected	0.20	0.60	0	0	0.05
Open room detected	0	0	0.90	0.10	0.15
Open passage detected	0	0	0.002	0.90	0.60

According to [12], an example given by Dervish, the original state set is given as {a-b, b-c}. The states have certainty factors of 1.0 and 0.2, respectively. The main problem that exists is how will the state-set progress if the robot concurrently detects an open passage on its left and an open door on its right? State {b-c} will progress to states c, c-d, and d. The states c and c-d can be ignored because the likelihood of detecting an open door when the actual feature is a wall is zero. The likelihood of being in state d (Y) is the product of

(1) The starting certainty factor for state b-c, i. e 0.2;

(2) The likelihood of not detecting anything at node c;

(3) The likelihood of detecting a passage on the left and a door on the right at node d.

The second likelihood (2) occurs only if it fails to detect the door on its left node c i.e [(0.6) (0.4) + (1 - 0.6) (.04)], also fails to detect open passage on right [(0.6) (0.4) + (0.4) (.001)] and correctly detects nothing on its right, .80. The third likelihood (3) occurs if it correctly identifies the open hallway on its left at node d, .90, and detects nothing on right .80.

The final formula which determines the likelihood or certainty factor for being in state d is as given in equation 13.

$$Y = (0.2)\,[(0.6)\,(0.4) + (0.4)\,(0.04)]\,[[(0.6)\,(0.4) + (0.4*.001)]\,[(0.9)\,(.80)] \quad \ldots\ldots (13)$$

$$= 0.008621$$

State a-b will potentially progress to states b, b-c, c, c-d, and d. Again, states b-c, c, and c-d can all be eliminated because the likelihood of detecting an open door when a wall is present is zero. The likelihood for state b is the product of the initial certainty, (1.0); the likelihood of detecting the door on its right as an open door, [(0.6) (0) + (0.4) (0.9)]; and the likelihood of correctly detecting an open hallway on its left, 0.9.

The certainty factor (X) for being at state b is given as in equation 14.

$$X= (1.0)\,(0.4)\,(0.9)\,(0.9) \quad \ldots\ldots (14)$$

$$= 0.324.$$

According the values in expression (13) and (14) it would then believe strongly that the robot is likely to be in state b but preserve the distant likelihood that it is in state d.

VIII. CONCLUSION

The robot used for the experimental purpose can successfully identify the landmark within a certain accuracy range. The main focus of the study is the navigation of robot and hence generation and storage of path map using cloud service. The master robot is the path creator and the follower request the path generated by master robot for it to follow the path. The follower robot needs to localize itself relative to the given map for which Markov localization is used. Calibration with respect to the difference in wheel is also depicted in the documents. But sometimes there are difficulties in controlling robots motion as it depends on sensors, feedback of controllers and internet connections. Many a times cloud based applications can get slow due to high latency responses or network problem. If robot is totally dependent on cloud a small fault in network can leave the robot brainless. But in future, robots can be made to store the interactions made with cloud so that if the connection goes offline, the robots can retrieve the stored data from their own memory in order to sense and react to their surroundings. Moreover the robots are heterogeneous in nature and masking of path is required for every different types of robot to understand and follow the master. But the research undergone does not consider the huge number of paths and in future it can be extended further to consider the situation and handle using the concepts of Hadoop Map reduce.

REFERENCES

[1] Machine learning, https://machinelearning.com supervised unsupervised, machine-learning-algorithm, Accessed on 22.02.17

[2] cloud machine learning platform/ https://cloud.google.com/ machine-learning, Accessed on 21.02.17.

[3] Landmark detection with the Cloud Vision AP, https:// cloud.google.com big-data, Accessed on 14.02.17

[4] Scene Text Detection, Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882 Volume 4, Issue 5, May 2015

[5] Mrumal.K.Pathak, Javed Khan, Aarushi Koul, Reshma Kalane, Raunak Varshney, “ Robot Control Design Using Android Smartphone”,Information Technology, Savitribai Phule Pune University.

[6] Hayet, J. B., Lerasle, F., & Devy, M.: A visual landmark framework for mobile robot navigation. Image and Vision Computing, 25(8), 1341-1351.(2007)

[7] Zhang, S., Sun, L., Dong, L., Yuan, B., Chen, Z., & Liu, J.: Initial location calibration of home service robot based on 2-dimensional codes landmarks. In Control Conference (CCC), 2014 33rd Chinese (pp. 8315-8320). IEEE.(2014)

[8] Lee, S. J., Lim, J., Tewolde, G., & Kwon, J.: Autonomous tour guide robot by using ultrasonic range sensors and QR code recognition in indoor environment. In Electro/Information Technology (EIT), IEEE International Conference on (pp. 410-415). (2014)

[9] Zainuddin, N. A., Mustafah, Y. M., Shawgi, Y. A. M., & Rashid, N. K. A. M.: Autonomous navigation of mobile robot using Kinect sensor. In Computer and Communication Engineering (ICCCE),International Conference on (pp. 28-31). IEEE.(2014)

[10] Limosani, R., Manzi, A., Fiorini, L., Cavallo, F., & Dario, P.: Enabling global robot navigation based on a cloud robotics approach. International Journal of Social Robotics, 8(3), 371-380.(2016)

[11] Dieter Fox, “Markov Localization: A probabilistic framework for mobile robot localization and navigation” University of Bonn, Germany, Doctoral thesis, 1998 pp 21-25

[12] Roland Siegwart “Autonomous mobile robots” ISBN-978-81-203-4322-1, MIT press,Cambridge,MA,U.S.A

[13] Kurfess, T. R. (Ed.). : Robotics and automation handbook. CRC press.(2004)