

Localization and Visual Navigation of a Scalable Robot Swarm using ArUco Markers

by

Name	Roll No.	Registration No:
Swarnendu Sardar	11700116026	161170110087 of 2016-2017
Sounak Mondal	11700116035	161170110078 of 2016-2017
Soumik Dhar	11700116039	161170110074 of 2016-2017
Arindam Samanta	11700116098	161170110015 of 2016-2017

A comprehensive project report has been submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology *in* **COMPUTER SCIENCE & ENGINEERING**

Under the supervision of

Dr. Pramit Ghosh
Assistant Professor
Dept. of Computer Science & Engineering
RCC Institute of Information Technology



Department of Computer Science & Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
Affiliated to Maulana Abul Kalam Azad University of Technology, WestBengal
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015

July, 2020

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY**



TO WHOM IT MAY CONCERN

I hereby recommend that the project entitled “**Localization and Visual Navigation of a Scalable Robot Swarm using ArUco Markers**” prepared under my supervision by Swarnendu Sardar (11700116026), Sounak Mondal (11700116035), Soumik Dhar (11700116039) and Arindam Samanta (11700116098) may be accepted in partial fulfillment of the requirements for B.Tech degree in **Computer Science & Engineering** from **Maulana Abul Kalam Azad University of Technology, West Bengal**.

.....
Project Supervisor
Department of Computer Science and Engineering
RCC Institute of Information Technology

Countersigned:

.....
Head
Department of Computer Sc. & Engg,
RCC Institute of Information Technology
Kolkata – 700015.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY**



CERTIFICATE OF APPROVAL

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR
EVALUATION OF PROJECT

1.

2.

3.

(Signature of Examiners)

ACKNOWLEDGEMENT

We would like to take this opportunity to express our profound gratitude and deep regards to our mentor – Dr. Pramit Ghosh who gave us the golden opportunity to work on this project and also for all his guidance, mentoring and constant encouragement throughout the course of this project.

We would also like to extend our gratitude to all the people for their direct or indirect involvement towards the completion of this project. Also, not to forget the coordinated efforts put forward by our work and hereby we thank each other for the successful completion of the documentation.

Arindam Samanta (CSE2016/081)

Sounak Mondal (CSE2016/083)

Soumik Dhar (CSE2016/084)

Swarnendu Sardar (CSE2016/087)

Abstract: This project deals with one of the most interesting problem in the field of robotics - localization and navigation of a mobile robot. The ability of a robot to find its location in the form of x and y coordinates in the 3D space or effectively in the 2D space and then finding its way to reach the destination is a very common problem and is also a very computation intensive and complex task. It has to take into account not just its own position relative to a reference point in the 2D space but also calculate the distance between the destination and its current position, as well as do the same for any obstacles that might come in its way. Not to mention the great applications that they provide, autonomous robot navigation can be used from regular household works to heavy duty industrial work.

Keywords: Localization, Navigation, Indoor Navigation Systems, Fiducial Markers, ArUco, Swarm Robotics

TABLE OF CONTENTS

CERTIFICATE OF APPROVAL.....	3
ABSTRACT.....	5
CONTENTS.....	6
LIST OF FIGURES.....	7-8
LIST OF ABBREVIATIONS.....	9
 <u>CHAPTER-1</u>	
INTRODUCTION.....	10
LITERATURE REVIEW.....	11-13
 <u>CHAPTER-2</u>	
SYSTEM DESIGN.....	14-15
METHODOLOGIES OF IMPLEMENTATION.....	16
SOFTWARE & HARDWARE REQUIREMENTS.....	17-18
OUR APPROACH.....	19
 <u>CHAPTER-3</u>	
IMPLEMENTATION DETAILS.....	20-34
SYSTEM VALIDATION.....	35
OBSERVED OUTPUT.....	36-41
RESULT DISCUSSION AND ANALYSIS.....	42
FUTURE SCOPE.....	43-44
CONCLUSION.....	45
REFERENCES.....	46
APPENDIX (SOURCE CODE)	47-50

LIST OF FIGURES

Figure No.	Title	Page No.
Fig 2.1	High level abstraction of data and instruction flow among different interacting components within the project	14
Fig 2.2	Comprehensive view of work-flow and data exchange among different interacting components within the project	15
Fig 2.3	ArUco-Gen web app for creating ArUco markers with fixed and random IDs	17
Fig 2.4	2WD robot chassis with Arduino Nano, L298N motor driver, BO-motors for drive and 3.7V 1500mAh Li-ion cells for powering the bot	18
Fig 2.5	Bolt IoT Wi-Fi module with ESP8266-12E	18
Fig 3.1	Landing page of the ArUco - Gen web application	20
Fig 3.2	ArUco marker generation based on fixed or random IDs [with provisions for selecting marker size and units (px, mm, inches)]	21
Fig 3.3	Generating multiple ArUco markers at a time (up to 500 markers allowed)	21
Fig 3.4	An ArUco marker with an ID of 358 and size of 400px X 400px	22
Fig 3.5	Barrel and Pincushion distortion in a standard chessboard	23
Fig 3.6	7x6 test pattern drawn from left-to-right, top-to-bottom	24
Fig 3.7	Using either cv2.undistort() or remapping to undistort the image	25
Fig 3.8	An ArUco marker with a green bounding box and three axes drawn at the center [shown using three different colors - x: red, y: green, z: blue (pointing outwards)]	26
Fig 3.9	ArUco Scanner Android application (server running at 192.168.2.2:5000)	27
Fig 3.10	Five random ArUco markers detected using the ArUco Scanner app	28
Fig 3.11	Pose estimation of an ArUco marker	29
Fig 3.12	Marker orientation of a cube of ArUco markers	30
Fig 3.13	Scanned output and simulated localization	30
Fig 3.14	ArUco marker localization setup	31
Fig 3.15	Ten random ArUco markers generated using the ArUco - Gen web application	32
Fig 3.16	Bounding boxes and heading lines generated for the ten markers using the ArUco Scanner Android application	32
Fig 3.17	Pygame window with plot of detected markers - bounding box (in green), heading line (in cyan) and ID (in white). Roll, pitch and yaw of the markers are logged in the terminal	32

Fig 3.18	A consolidated representation of the localization process	33
Fig 3.19	A test simulation showing a robot, two barrier objects and a target	34
Fig 3.20	A consolidated representation of the navigation process showing the ArUco markers, marker attitude, and simulation.	34
Fig 3.21	System validation [from top, ArUco-Gen (generator), ArUco Scanner (detector) and ArUco Scanner Nav Sim (localization and navigation simulator)]	35
Fig 3.22	A simple localization and navigation setup using ArUco Scanner logged on the laptop screen	36
Fig 3.23	Output - time-stamp: 00:01:25; frame no.: 5	37
Fig 3.24	Output - time-stamp: 00:03:75; frame no.: 15	38
Fig 3.25	Output - time-stamp: 00:06:25; frame no.: 25	39
Fig 3.26	Output - time-stamp: 00:11:25; frame no.: 45	40
Fig 3.27	Output - time-stamp: 00:13:75; frame no.: 55	41
Fig 3.28	Intended simulation of way-point navigation of a swarm of robots with barriers and objects	43
Fig 3.29	A simple 2WD robot build using Arduino Nano and L298N	44
Fig 3.30	A swarm robot configuration	44

LIST OF ABBREVIATIONS

ArUco:	Augmented Reality University of Cordoba
Bot:	Robot
Nav:	Navigation
OpenCV:	Open Source Computer Vision library
VBL:	Vision Based Localization
MQTT:	Message Queuing Telemetry Transport
GPS:	Global Positioning System
TCP:	Transport Control Protocol
PID:	Proportional, Integral, Derivative
Contrib:	Contributor
2WD:	2-wheel drive
IoT:	Internet of Things
BO-motors:	Battery Operation motors
Li-ion:	Lithium-ion
Json:	JavaScript Object Notation
Lib:	Library
Rad:	Radian
AR-marker:	ArUco Marker
px:	Pixel
rvec:	Rotation vector
tvec:	Translation vector