GIA-Cargo Team Description Paper

Amin Arriaga, Angel Garces, Carlos Perez, Eduardo Lopez, Wilfredo Graterol
Grupo de Inteligencia Artificial (GIA), Department of Computer Science and Information Technology,
Universidad Simón Bolívar, Venezuela

***Abstract*— This paper presents the basic concepts of the operation the robot designed and built by GIA-Cargo team to fulfill the tasks described in the rules of Latin American Robotics Competition IEEE Open. In this sense, it describes in detail the strategy to be used, the mechanisms designed, the components necessary for its construction and the implemented algorithms.**

I. INTRODUCTION

The shipping industry is one of the most important pillars of global trade; in order to reduce delays in the departure of ships the automation cargo ports operations is a necessary and a key factor for a faster and more efficient trade industry.For this reason, the challenge of Latin American Robotics Competition (LARC) IEEE Open consists of managing a cargo port operations using an autonomous robot capable of handling containers with speed and precision.

To complete the objectives proposed, an autonomous robot is developed to perform the tasks of: locating the metal blocks simulating the shipment containers, identifying the blocks color and moving and placing the blocks in the designated areas for each type of block. To refine each aspect of the robot development, a test field is constructed with the measures specified in the rules of competition.

II. STRATEGY

The cargo area where the blocks are located at the beginning is divided into three joint areas bounded by black lines and surrounded by a green square. We will refer to the middle area as the central garden, to the other two as lateral gardens. The robot will focus on taking the blocks of the lateral gardens to their corresponding ships or in the case of a red block, in the train, leaving the blocks of the central garden as the last.

At the beginning of the route, the robot is in an area in the middle of the ships in front of the central garden, where there isn't a line to guide himself, so it will simply advance until it reaches any line. Once you reach the first line, corresponding to the green line, you will turn to be able to follow it. The orientation of the turn will initially be in a clockwise direction, as it will be directed to the blocks in the right field.

Then, the robot crosses the green line until finding a block to its laterals, being parallel to this one, in this moment it advances a little, turns and advances again, to be able to take the block, taking advantage of the delimitation of the garden to line up next to the block. Then, depending on the color, the robot will make a different decision.

If the block is green or blue, the robot will return to the green line and follow it in the opposite direction until reaching the central garden. To achieve that, it will detect the position of the blocks on its sides. At this moment, he will position himself between the ships and after verify the color of the block, to determine in which area should place it.

If the block color is red, the robot will return to the green line and will continue in the direction that came, until he can't find a line on the floor. At this moment the robot will advance until it finds something in front of it, and will look from right to left for any free position on the train to leave the block.

After this, he will be able to repeat this process until he take all the blocks of the right garden to their corresponding zones, and then do the same in the left field.

After carrying all the blocks of the lateral gardens, it will be concentrated in the central garden following the same strategy, with the difference that when it detects the blocks from the exterior lines, it will advance until it reaches the delimitation of the central garden, it will proceed to take the block until its corresponding area.

III. MECHANICS

A. Structure (Dimension and Materials)

In order to be stay structurally stable and lightweight the robot main structure will be made of 0.13cm thick. aluminum sheets.

B. Crane

Due to the blocks being made of steel; the crane of the robot will consist of an electromagnet being held in place by two bamboo skewers, and connected using nylon cord to a 32mm radius Lego wheel that will be attached to a gear reduction system made using Lego pieces powered by two Lego Dc 9V Mini-Motors (71427c01) [1]. There will be also an HC-SR04 Ultrasonic sensor [2] at the bottom of the crane to measure the height of the electromagnet in any given moment.

Lego pieces are used in various mechanisms related to the crane because these are readily available for the team, are considerably easy to ensemble and provide the necessary structural strength to support the weight of the electromagnet together a metal block.

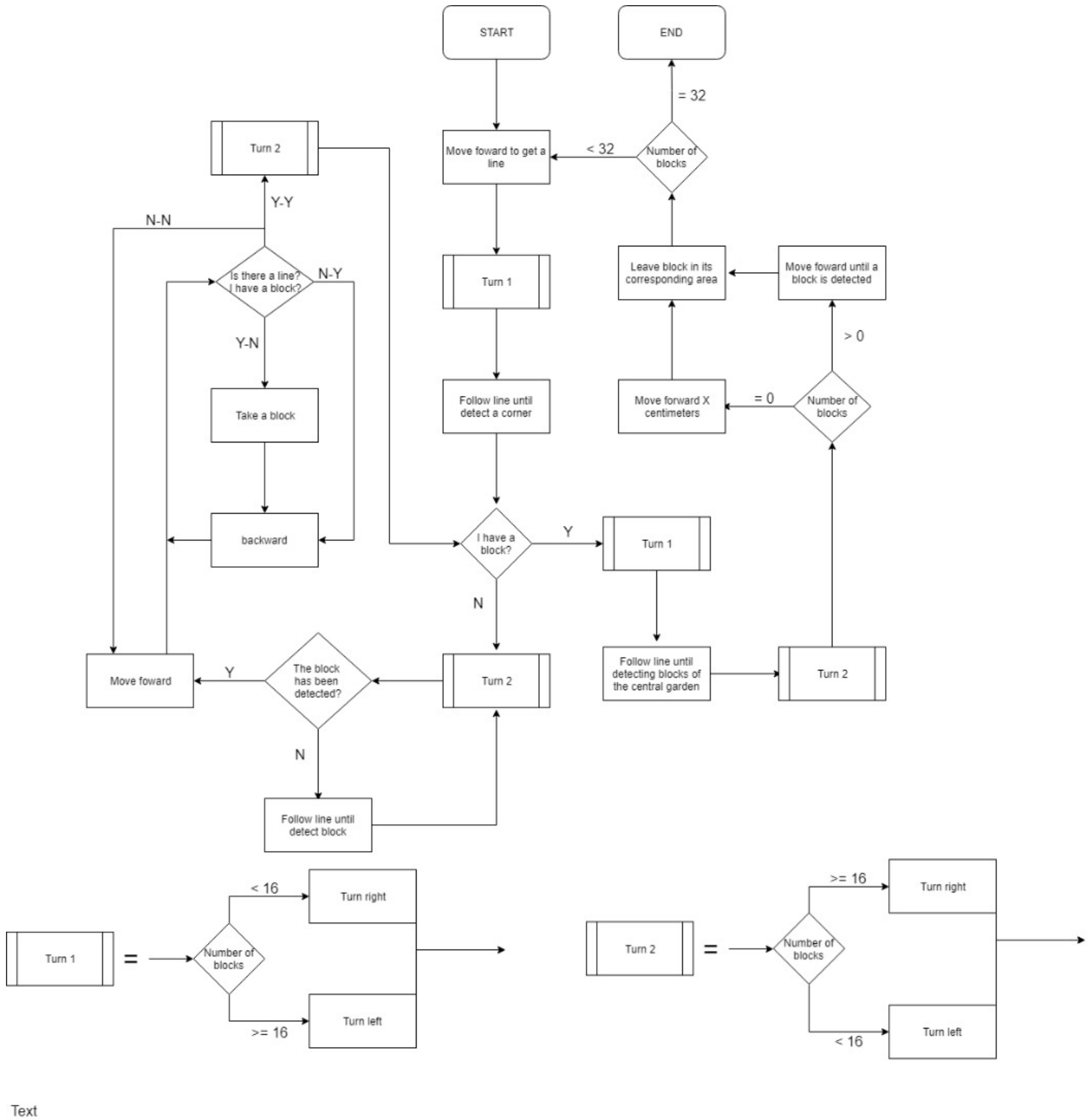

Fig. 1. Startegy flow chart

C. Mobility

The robot will be provided with two 40.8mm radius Lego wheels, these will provide sufficient traction for the course even terrain. The wheels will each be attached to the Pololu Gearmotor LP[3] with gear reduction boxes of 75:1 relation, being able to move a weight of 5000g, increasing control and allowing a linear velocity of around 49 cm/s.

IV. HARDWARE

The single board computer Raspberry Pi 3 Model B [4], was chosen as the robot main processing unit, the Raspberry will be used as the main interface to interact with the robot actuators and the majority of the sensors which include; the Ultrasonic HC-SR04 sensors [2] for measuring distance, the two encoders attached two the Pololu motors, an on and off switch located on the top of the crane.

Along with the raspberry an Arduino Duemilanove microcontroller [5] is used to process analog signals from a QTR reflectance sensor [6], and is connected to the raspberry using a USB type A to OTG cable, for communication and for powering the arduino.

The camera selected to perform the image computer for the robot is the Raspberry Pi Camera Module v1 [7], with an OmniVision OV5647 sensor, it is perfectly fitted to work with the raspberry pi 3; It can take pictures up to 5 Megapixels of resolution and record 1080p30 resolution video. These

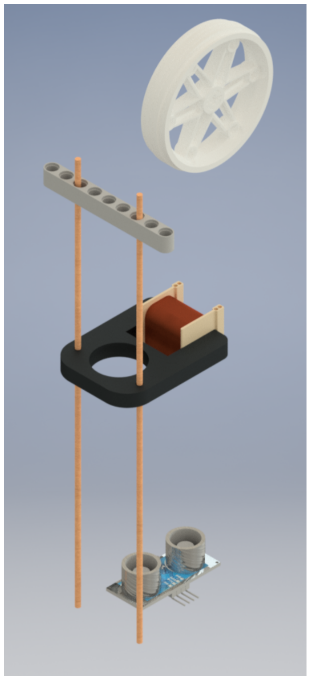

Fig. 2. Crane Model

features are more that enough to recognize in real time the shape and color of the blocks on the field.

To control the the electromagnet and the crane motors a dual H-bridge us used; also for the pololu gear-motors used for the robot movement the Dual MC33926 Motor Driver Carrier [8] is used to regulate the current output to the motors using a pwm signal.

The raspberry and Ultrasonic sensors are powered by a power bank that provides a 5V output. All of the robot Dc motors (mentioned in section III), and the electromagnet are powered by a 11.1 V 5500 mApH Lipo Battery using circuit to regulate the output voltage to each component.

V. ALGORITHMS

A. Vision

1) Color Recognition: As explained in the competition rules, there are 3 specific colors we need to recognize, not only on the containers but on the delivery zones.

Color recognition is achieved by manipulating each pixel, using its RGB values. RGB stands for Red Green Blue. The library we'll be using for such work is opencv, more specifically its Python distribution 'opencv-python'.

We want to eliminate noise from the image in order to get the clearest and smoothest color we can, so we can detect a consistent surface on the image. To obtain this result we'll be using a Gaussian filter. This method of filtering is similar to a mean filter (replacing the value with a mean of its neighbors) but uses a different kernel that resembles a Gaussian bell.

The idea is that the matrix representing the surrounding neighbors of a given pixel has sort of a spreading weight over the center pixel. The closer it is to the center the more weight has over the final result.

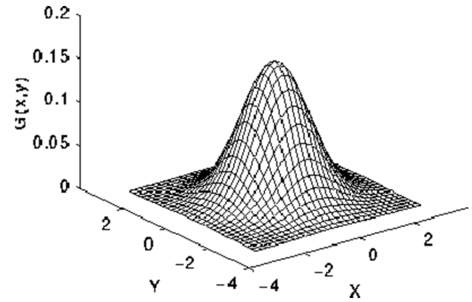

Fig. 3. 2D Gaussian distribution

After the image has been blurred we proceed to convert the representation of the image to HSV, meaning Hue Saturation Value, which separates the color intensity from the color information so we can focus on color and its segmentation.

We define the filters for each color, each color bounds, using the min and max values of each of the colors that we'll identify in the images. With this we'll create a mask or filter so we can focus on the color and eliminate the rest. We define a second mask to filter by intensity of the brightness. Finally we combine the masks.

With this mask we are going to highlight the resulting structure on the image. We highlight only the biggest structure we find, being the closest to the cart. We overlay the mask to the image.

Finally we convert back to the original color scheme and this way we get the image with the biggest container, being the closest, highlighted. We will use this method to determine not only the best possible container to pick but to know where to take it to.

B. Navigation System

To move the robot between the areas we will be based in the different colors of the ships area and the cargo area (blue and green respectively), also there are a black line inside the cargo area. which allow the robot to know until what time stop moving and search a block.

Following the idea above, there have been designed and implemented four different controllers algorithms: followLine, keepGoingUntilShip, followLineUntilBlock and reachTrain.

followLine: This code uses a PID controller to make the robot follow the line, he start sending the same amount of voltage in each motor, and after pass dt = 0.005 seconds, calculate the error with respect to the line using an average between the values of the eight sensors, and he uses the following formula (where kp, ki, kd are constants) to calculate the value added to the left or right motor.

$$\text{porportional} = \text{average} - \text{center value} \tag{i}$$

$$\text{integral} = (\text{integral} + \text{proportional}_{past}) \cdot dt \tag{ii}$$

$$\text{derivative} = \frac{1}{dt}(\text{proportional} - \text{proportional}_{past}) \tag{iii}$$

$$\text{PID}_{out} = kp \cdot \text{proportional} + ki \cdot \text{integral} + kd \cdot \text{derivative} \tag{iv}$$

keepGoingUntilLine: Same as the previous method, uses a PID controller but, this method calculate an error respect the difference between the enconders values and use 0 as center value, and ends when any of the sensors have value greater than 500 (an upper bound of the white reflectance value)

followLineUntilBlock: The purpose of this method is follow the green line until a block is found. To do this, he call the method followLine until he found in any of the lateral ultrasonic sensors a distance lower than 20 cm. The robot advances 5 cm and he will search a block.

reachTrain: It is a method used by the robot to reach the train, first use a PID to follow a line until he finds a value lower than 500 in all the sensors, after that he advance (with another PID but taking as reference the different between the encoders) until finds an distance lower than 10 cm on the ultrasonic sensor which is in front of the robot.

He need differentiate the areas, to do that he use his reflectance sensors to distinguish a change of color on the floor. Also use the Arduino Library for the Pololu QTR Reflectance Sensors [6], to know the level of reflectance of the floor.

In the cargo area the robot will follow the green line with a PID controller. Taking as reference the position of the line in the eight sensors. And when the robot wants to leave a block, he positions himself in front of the ships and go forward until he finds a blue line, using the method keepGoingUntilLine.

C. Blocks interaction

It's necessary to the robot be able to leave and get a block from a stack, for that, he use a function and a method: grabABlock and leaveBlock.

grabABlock it's a function that returns the color of the block and get this block with the magnet. He use the same strategy of the method keepGoingUntilLine where's calculated a PID to advance until he reach a distance with respect to the blocks lower than 0.5 cm. In this moment the robot use the camera to know in which of the four levels from the top to bottom there's a block and its color, and depending of this, the magnet goes up and then it turns on.

leaveBlock is a method used to leave the block in the ships area, he places the magnet in the maximum position and descends to a level above the highest block detected by the camera, and then turns off the magnet.

When the robot is in front of the blocks, he use the method keepGoingUntilLine until to reach the black line, after that he use the function grabABlock, and for leave the block on the ship, if the block is green or blue, he will positions himself in the middle of the ships and decide which side to turn depending on the color of the block, to then use the method keepGoingUntilLine followed by the method leaveBlock. If the block is red, the robot. will return to the green line and use the method reachTrain followed by the method leaveBlock if robot finds an empty wagon.

VI. Conclusion

Due to its simple design, use of low-Cost materials and implementation of computer vision algorithms the robots can be easily maintained and improved over time in order to replicate and to scale up this model to be implemented as viable solution to the problem of cargo port management.

Acknowledgments

Team members are deeply grateful with their families for believing in this project and for everyday support. Also, the team wants to extend its gratitude to the Universidad Simón Bolívar, the Department of Computer Science and Information Technology, as well as professors Carolina Chang and Carolina Martínez Acknowledgments to David Henandez, Maria Gabriela Guzman, Lalezka Duque, Eliot Díaz, Arturo Toro, Anthony E Kareh Marcos Jota, Catherine Lollett, and Raúl Bander, fundamental GIA's members and unconditional friends.

References

[1] Lego Dc 9V Mini-Motors (71427c01) [Online]. Available: https://www.bricklink.com/v2/catalog /catalog-item.page?P=71427c01#T=C

[2] Ultrasonic Ranging Module HC - SR04 [Online]. Available: http://www.micropik.com/PDF/HCSR04.pdf

[3] Pololu Gearmotor LP [Online]. Avilable:https://www.pololu.com/product/2275

[4] Raspberry Pi 3 Model B [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf

[5] Arduino Duemilanove [Online].Avilable:https://www.arduino.cc/en/Main/arduinoBoardDuemilanove

[6] Pololu Corporation. (2001). Arduino Library for the Pololu QTR Reflectance Sensors [Online]. Available: https://www.pololu.com/docs/0J19/all

[7] Camera Module [Online]. Available: https://www.raspberrypi.org/ documentation/hardware/camera/README.md

[8] Dual MC33926 Motor Driver Carrier [Online]. Available: https://www.pololu.com/product/1213