Zone-sensitive RIZOBots in Action: Examining the Behavior of Mobile Robots in a Heterogeneous Environment

Ozan Balcı[1], Sema Alaçam[2]

[1] Istanbul Technical University, Istanbul, Turkey
balcioza@itu.edu.tr; alacams@itu.edu.tr

Abstract. This study proposes a framework for the use of mobile robots namely RIZOBots in form studies in the early phases of design. The proposed framework was tested in two experiments. An agent-based model was utilized for the movement of mobile robots, a drawing task was defined as the task. In particular, rule sets for agent-agent and agent-environment interaction were used. Light-sensitivity rules were utilized to achieve agent-environment interaction, apart from obstacle detection. This study focuses on the effects of two different zone-related states on the behavior of RIZOBot which is a configurable differential-drive wheeled robot developed by authors using off-the-shelf products and 3D printed body parts. Two zone types with very basic features are used to define environmental conditions. The traces left on the canvas, the irregularities in the movement of the robots, and the robot-environment interaction will be evaluated in the study. The results and analysis of the two selected experiments are presented and the potential of the proposed framework is discussed.

Keywords: Robotics, Swarm robotics, Swarm behavior, Mobile agents, Zone-sensitivity

1 Introduction

When Negroponte's (1970) "The Architecture Machine" first published, it was clearly stated that the technology of the machines that were theoritized at that time was not available. Yet, Negroponte (1970) introduced three types of interaction regarding the involvement of machines in architectural design processes. These three types are as follows: Speeding up the already existing procedures through automation, combination and alteration of existing methods, and symbiotic interaction between designer and machine that can evolve and become mutually transformative through time (Negroponte, 1970). With Negroponte's (1970) taxonomy, this study can be considered as an attempt to stretch the second type towards the third through low-cost devices and existing technologies. In this sense, apart from the experiment scenarios, the rules and relations defined by the authors, can be taken as the original contribution of this stretching attempt. Another issue that Negroponte (1970) emphasizes is the context-awareness. With Negroponte's (1970) words, "*it must have a sophisticated set of sensors, effectors, and processors to view the real worlds directly and indirectly*". Therefore, the architecture machine is expected to recognize the context and process the information before the actualization of an operation. Keeping Negroponte's promises in mind, this work introduces a proof-of-concept study of a design exploration space involving multiple robots.

In this study, we present a framework to produce generative drawings with an autonomous and decentralized multi-robot system. The components of the proposed framework are identical multi-robots namely RIZOBots, a physical environment as an experiment setup, and ongoing interaction between multi-robots and the environment. Firstly, we reconfigured the mobile robots named RIZOBot. RIZOBot is a custom-made wheeled robot developed by the authors to implement diverse studies on the topic of swarm systems. RIZOBot has a core that allows customizing by adding various sensors. In this study, we used light sensors to detect the zones created by different light intensities, and distance sensors to detect obstacles.

The proposed framework is tested within a series of experiments. The experiment setup covers four identical mobile robots equipped with different colored markers, a white canvas as an environment, multiple light sources to create zones and an action camera to record the process. Specifically, we investigate the traces of mobile swarm agents to understand how they interact with other agents and zones with various features.

2 Swarming: From Simulation Towards Actualization

Swarm systems have been studied by many scholars to describe self-organized and complex systems behaviors with bottom-up simple rules and relations (Reynolds, 1987; Theraulaz et al., 1990; Brooks, 1991; Deneubourg et al., 1992; Johnson, 2002). In a broader sense, a system based on swarm behavior is expected to perform more complex and efficient than the arithmetic sum of parts. Basically, this complexity results from the internal dynamics of the system, the interaction of its parts with each other and with the environment. Deneubourg et al. (1992) define swarm as:

"*a set of (mobile) agents which are liable to communicate directly or indirectly (by acting on their local environment) with each other, and which collectively carry out a distributed problem solving*" (Theraulaz et al., 1990; Deneubourg et al., 1992:123).

This conceptualization of swarm by Deneubourg et al., (1992) contributes to the representability of a swarm system through computational models that cover agents, tasks, functions, and relations. In other words, without a need to direct interaction, a system consisting of agents, the interaction between agent-agent, agent-environment, and agent-action enables models for stigmergic mechanisms (Deneubourg et al., 1992). Reynold (1987) has introduced more specific elements namely ‘collision avoidance’, ‘velocity matching’, and ‘flock centering’. Further to the Reynold’s (1987) assumption, the alignment-cohesion-separation structure has become foundational functions for many algorithms in the swarm literature. Different from a vast body of studies in the literature that focus on representing swarm behavior through digital models, this study aims to provide a better understanding of the behavior of swarm systems in real-world scenarios.

Brooks (1991) puts a special emphasis on real-world scenarios, ‘real sensing’, and ‘real action’ cases regarding the development of intelligent systems. Brooks' proposition, which is nurtured from Heidegger's "being-in the world" discussion, makes it interesting to test models of swarm behavior, which are abstract representations of the real world, in real-world scenarios. Moreover, considering swarm and robotics together, Beni (2005) highlights the potential of key concepts and qualities such as robustness, modularity, scalability, and interchangeability. Discussing the concept of intelligence in the context of swarm robotics, Beni (2005) uses the term "ordered pattern, improbable outcome". While soft components of a robotic system such as programming enable ordered patterns to be achieved, the mechanical parts of robots involved in real-world scenarios trigger improbable outcomes. In this context, Beni (2005) introduces a definition of “intelligent swarm” as follows:

“*a group of non-intelligent robots forming, as a group, an intelligent robot. In other words, a group of “machines” capable of forming “ordered” material patterns “unpredictably*” (Beni, 2005: 4).

There are remarkable studies that utilize multi-robot systems in architectural form explorations (Andreen et al., 2016; Petersen & Nagpal, 2017; Bao et al., 2021). Different from Andreen et al.’s (2016) use of mini robots with a simple movement behavior, this study employs 2 different sensors to inform the mini robots. Secondly, predefined route paths are used in the implementation part of this study. Third, the light quality of the environment in the experiment setup provides a changing feature that influences the complexity of the robot’s behavior. Finally, the mechanical part of the robot body differentiates from the existing studies. Bao et al. (2021) focus on the implementation of swarm logic in the optimization problems simulated in the digital environment. They integrate swarm-based generative design approach and structural topology optimization in their model (Bao et al., 2021) without a need to realize and test in real-world conditions. Petersen, & Nagpal (2017) present a review of real-world implementation of multiple robots in relation to design and construction problems. Besides the aforementioned studies, there are many studies dealing with swarm robotics in relation to the the context of art and design (Valentini et al., 2017; Dorigo et al., 2021).

3 Methodology & Implementation

In the scope of the implementation, two tests lasting 10 minutes were conducted. Each experiment setup consists of a 150*250 cm white canvas as the environment, two light sources with different light intensities and four identical mobile robots equipped with different colored markers. We limited the white canvas with 50 cm height fabric to prevent the mobile robots from leaving the experiment area, and we used black fabric in order to avoid light reflections. As light sources, we used one 100-lumen and one 50-lumen LED lamp hung 120 cm above the white canvas (Fig. 1). Since the light intensity is the main factor in our study, we conducted the experiments between 00.00-02.00 AM in a homogeneously illuminated indoor environment with approximately 20-lumen light intensity.

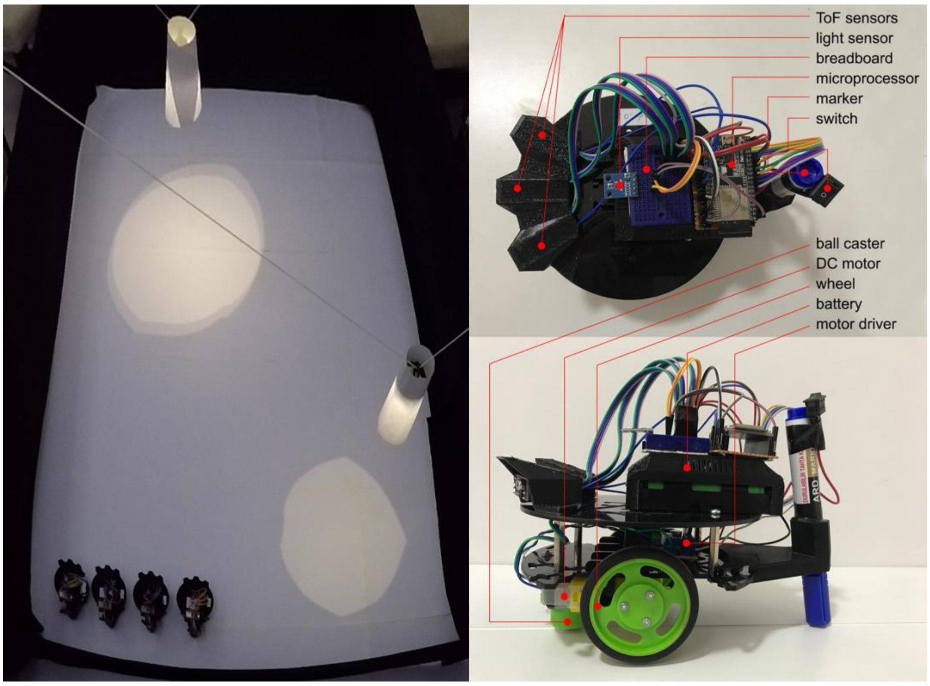

Figure 1. Experiment setup and RIZOBot Mini.

We used four RIZOBot Minis equipped with one of the black, blue, green and red markers as mobile robots to conduct the experiments.

3.1 Body

Each robot consists of 5 main units: chassis, sensors, processor, motor driver and battery unit. As chassis, RIZOBot has a plexiglass platform with 3D printed body parts, one ball caster, two wheels with 70 mm diameters and two 6V 250 RPM plastic geared DC motors which are connected to the wheels.

Robots are equipped with three VL53L0X time-of-flight (ToF) and one BH1750 light sensor. As is shown in Figure 1, one of the ToF sensors is positioned perpendicular to the direction of movement, other two sensors are positioned at an angle of 30 degrees to increase the range of the field of view. ESP32 Wroom-32D is the microprocessor used in the RIZOBots. The microprocessor runs the tasks according to sensor reads and powers up the DC motors via an L298N dual motor driver to create motion. Three series-connected 3.7V 18650 batteries are used as the power source for the whole system.

3.2 Soft Components

Instead of utilizing a central feedback mechanism for the multi-robot system, we programmed each robot individually with local rules to define behaviors for the agents. Three main behaviors of the robots are listed as follows: movement, obstacle avoidance, and light-sensing behavior. Robots can execute both linear and angular movements. Linear movement is obtained by rotating two motors at the same speed in the same direction, while angular movement is obtained by rotating them in the same direction at different speeds. In addition, robots can turn around themselves by rotating the motors in opposite directions at the same speeds. Obstacle avoidance behavior is utilized to avoid collisions among robots or the environment while robots are on the move. In order to accomplish obstacle avoidance, we used ToF sensors and operated them with Adafruit VL53L0X Library (2022). Obstacle avoidance has three different states: no obstacle ahead, obstacle ahead and obstacle on the sides. RIZOBot constantly reads the distances with middle and side sensors. If there is no obstacle within 300 mm, the robot continues its predefined movement path. Yet, if the robot encounters an obstacle within 300 mm directly in front of it, it avoids the obstacle by turning left or right around itself according to the readings it receives from the side sensors. In the third state if RIZOBot does not detect any objects within 300 mm directly in front of it, however, if it detects an obstacle within 300 mm by one of the side sensors, the robot avoids the obstacle by making an angular movement in the opposite direction. Obstacle avoidance behavior is provided (Fig. 2).

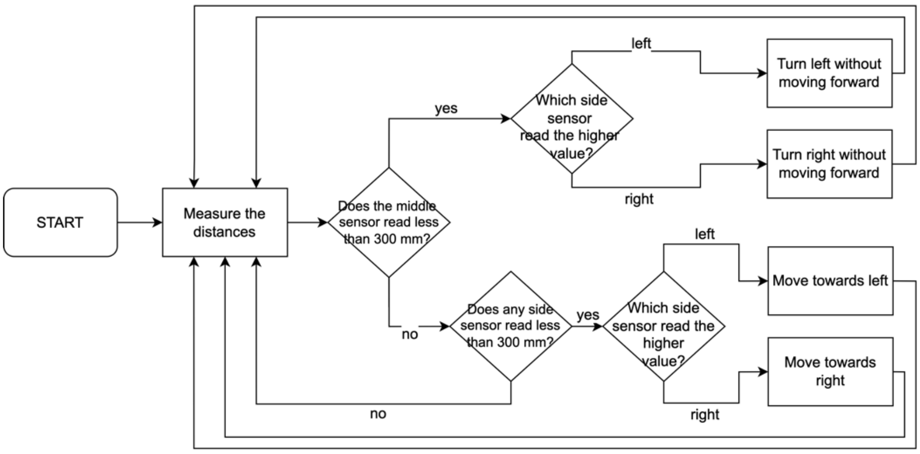

Figure 2. Flowchart of obstacle avoidance behavior.

Light-sensing behavior allows the robot to measure the light intensity at its location via using the light sensor with BH1750 Library (2022). Multiple light sources are utilized to create zones with intangible borders. In other words, the level of illuminance provided to encode a binary zone as dark and light. We developed a movement algorithm based on light intensity and designated it to mobile robots.

3.3 Experiment No.1

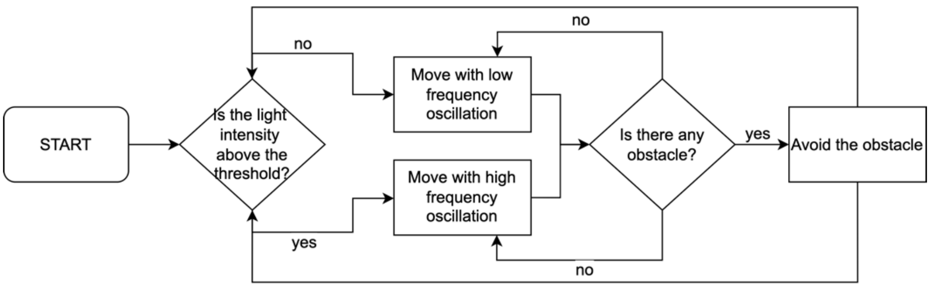

Figure 3. Algorithm schema for Experiment No.1

In the first experiment, we use two fixed position light sources to create two non-intersecting circle-like luminous zones on the white canvas which have light intensity higher than 45 lumens while the rest of the area has an average of 20 lumens. After we programmed the movement behavior of the RIZOBots in the Arduino environment to be a binary state dependent on light intensity. If the light sensor readings are below the predefined threshold, which is 35 lumens, the movement behavior enables the state no.1, and the robot moves forward (as illustrated in Fig. 4).

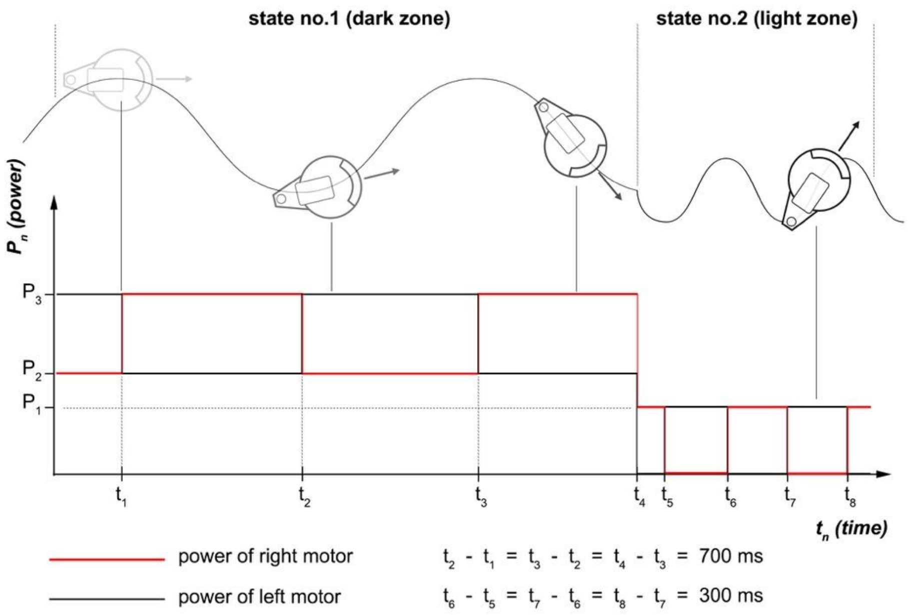

Figure 4. Movement states of RIZOBot.

In order to generate such a movement path, we use a time-integrated switch function to periodically change the power of the two motors every 700 milliseconds. A part of the code for the motion control is shared below:

```
switch = 0
timer = 0
period = 700 ms

function.stateOne()
if (switch = 0 AND (currentTime - timer) >= period)
leftMotor = P2
rightMotor = P3
switch = 1
timer = currentTime
if (switch = 1 AND (currentTime - timer) >= period)
leftMotor = P3
rightMotor = P2
switch = 0
timer = currentTime
End
```

Yet, if the light sensor readings are above the threshold, the mobile robot alters its movement path to state no.2 and increases the frequency of oscillation. The function used to generate the movement path in the first state is used in the second state for the same purpose only with different P1, P2 and period values (Fig. 4).

Marker-equipped four RIZOBots with obstacle avoidance, light-sensing and movement behavior with the aforementioned binary state run for 10 minutes and drew their traces on the white canvas.

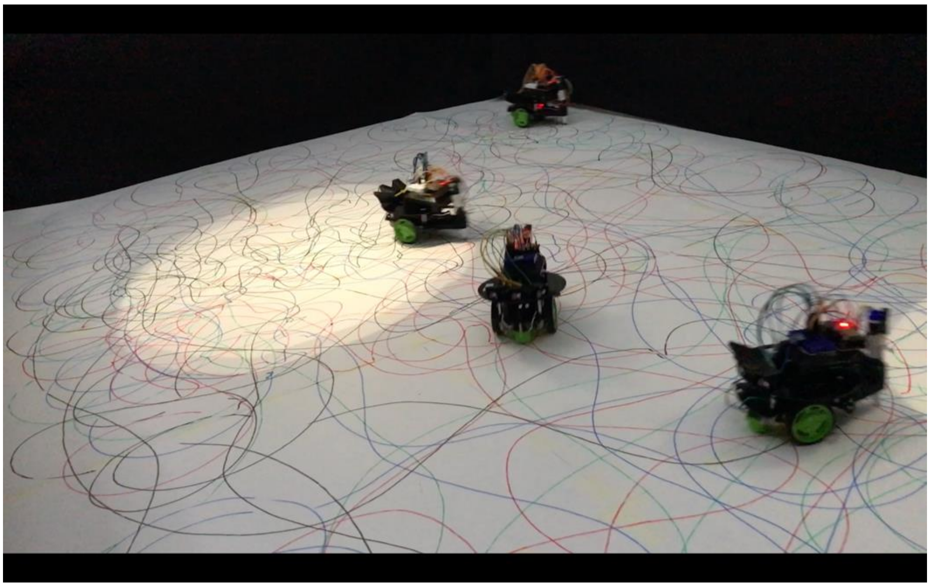

Figure 5. RIZOBots painting on canvas.

3.4 Experiment No.2

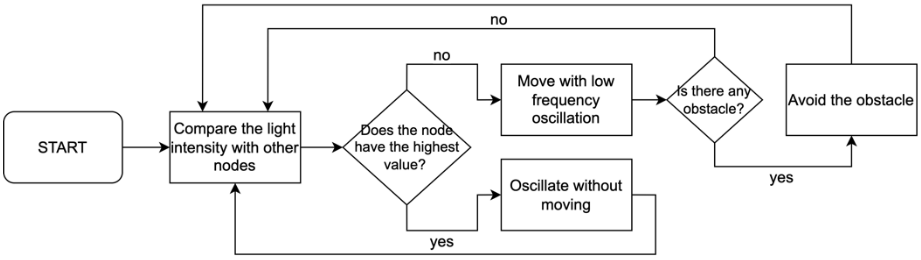

Figure 6. Algorithm schema for experiment no.2

As in the first experiment, we generated two non-intersecting circle-like zones using the same light sources on the white canvas. However, unlike in the first experiment, we manipulated and relocated the light sources throughout the experiment. Another major difference is that we have designated a communication behavior as well as the three behaviors that the RIZOBots have. RIZOBots use a Wi-Fi mesh network to communicate. Mesh network is a network topology where each agent may act like an access point and create a connection between all the nodes in different forms. We used the Painless Mesh Library (2022) to establish the mesh network in a star topology in which one or more agents become root nodes and the rest of the nodes connect to them. Since there is not any predefined hierarchy between the nodes, any node may become a root node depending on the position of the other nodes. After, we defined a function for RIZOBots to broadcast their light sensor readings, asynchronously, over this mesh network in JSON format by using Arduino JSON Library (2022). Every second, each robot shares the light intensity it measures with the rest of the group and compares it to determine the highest value. The robot with the highest value stops moving forward and draws semicircular shapes by oscillating around itself while other robots continue their movement in seek of a location with higher light intensity. If one of the other robots reads a higher value than the previous point, the robot switches roles with the robot which has the previous highest value and starts to do the oscillation movement while the other robot returns to the seeking state.

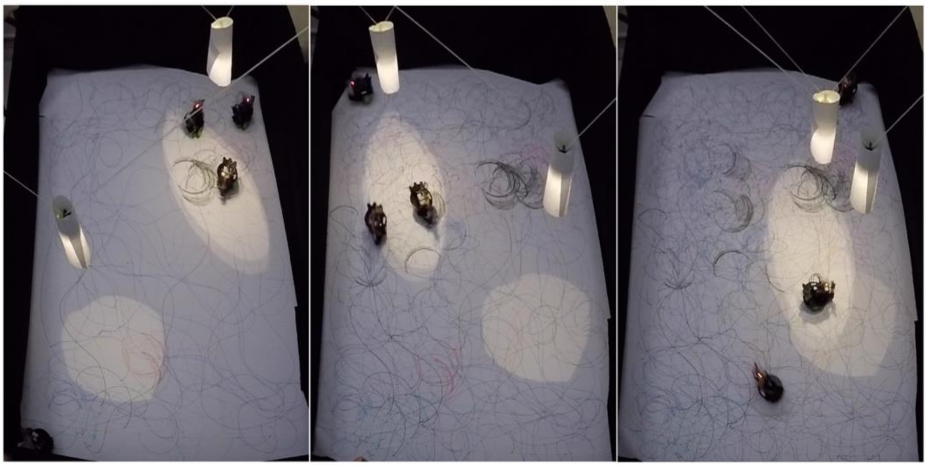

Figure 7. RIZOBots find the brightest point in different zone layouts.

We carried out the experiment with 3 RIZOBots due to the failure of one of the robots after the previous experiment.

3.5 Findings & Outcomes

- The proof-of-concept study has been conducted successfully for the proposed RIZOBot framework (Fig. 8).

- All four RIZOBots in both experiments have executed the given drawing task through the utilization of the defined algorithms. In addition, all robots accomplished remaining in the defined boundaries throughout the experiments.
- The interactions among the RIZOBots and between RIZOBots-environmental factors(light) have been monitored through the generated drawing patterns.
- In the first experiment, although the states of luminous and dark have been binary coded, during their movement the RIZOBots performed smooth transitions from one zone to another.
- Another gap that emerged in the intended algorithm and actualized behavior is obstacle avoidance. Some collisions could not be avoided. It is thought to be due to the structure of motors used.
- The movement mechanism of the robots is provided by giving power to the right and left wheels, sequentially. In this case, low precision results were obtained in rotation movements. The precision of movement can be increased if an intermediate layer such as an encoder is used to control the RPM values of the wheels.
- In the second experiment, it is observed that the highest light sensor value could have been detected in approximately 3 minutes time by the RIZOBots. The system as a whole responds to the changing light conditions as real-time feedback.

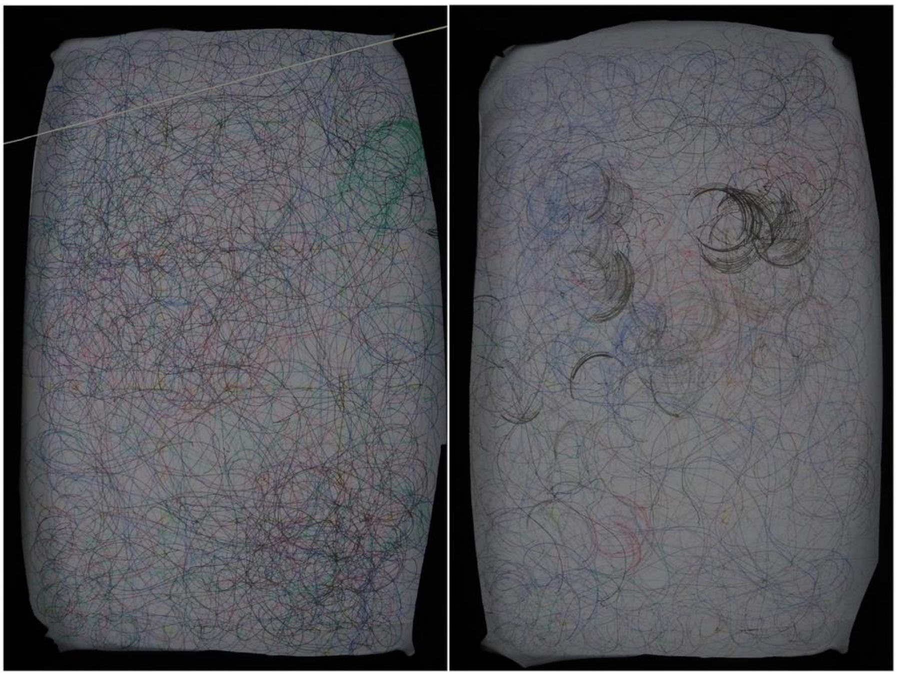

Figure 8. Experiment results (no.1 on left, no.2 on right).

4 Concluding Remarks

In this study, the potentials of swarm behavior are explored through experiments that involve multiple robots, predefined algorithms for movement, and actions defined for the changing light intensity of the environment. Relatedly, the ability to sense and respond to the changing light intensity of an environment by an individual robot, “zone-sensitivity”, has been tested.

During the experiments, we observed several collisions caused by high velocity and short delays in state changes due to latency in sensor readings. These minor errors aside, the framework successfully performed the given tasks in both experiments. When we examined the traces of the robots, we observe consistency in their reactions to different situations such as encountering boundaries, avoiding each other, and change of states due to the light intensity. Besides this consistency in the reactions, the communication behavior used in experiment no.2 provided a stable message transmission over the mesh network between the agents.

The results of the experiments offer that the proposed framework carries the potential to be adapted to different design and representation problems in future studies. More complex tasks can be achieved by defining larger amount and more complex behaviors to RIZOBots.

This study can be considered as an initial attempt to explore the potentials of interaction between a designer and a mechanical system to achieve emergent and unexpected results. Development of the zone-sensitivity behavior might contribute to the context-awareness of the robots.

5 References

Arduino JSON Library. (Version 0.1.0) [Arduino IDE Library]. Retrieved July 12, 2022. https://arduinojson.org/

Adafruit VL53L0X Library. (Version 1.2.0) [Arduino IDE Library]. Retrieved July 12, 2022. https://github.com/adafruit/Adafruit_VL53L0X

Andreen, D., Jenning, P., Napp, N., & Petersen, K. (2016). Emergent structures assembled by large swarms of simple robots. In the Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), Posthuman Frontiers: Data, Designers, and Cognitive Machines, Ann Arbor 27-29 October, 2016, pp. 54-6, ISBN: 978-0-692-77095-5. https://doi.org/10.52842/conf.acadia.2016.054

Bao, D. W., Yan, X., Snooks, R., & Xie, Y. M. (2021). SwarmBESO: Multi-agent and evolutionary computational design based on the principles of structural performance.

Beni, G. (2005). From swarm intelligence to swarm robotics. In Swarm Robotics, LNCS 3342, pp. 1–9, 2005. Springer.

Brooks, R.A. (1991). Intelligence without representation. Artificial intelligence, 47(1-3), 139-159.

Deneubourg, J. L., Theraulaz, G., & Beckers, R. (1992). Swarm-Made Architectures. Varela, F. J., & Bourgine, P. (Eds.). Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life. MIT press.

Dorigo, M., Theraulaz, G., & Trianni, V. (2020). Reflections on the future of swarm robotics. Science Robotics, 5(49), eabe4385.

Johnson, S. (2002). Emergence: The connected lives of ants, brains, cities, and software. Simon and Schuster.

Laws, C. BH1750 Library. (Version 1.2.0) [Arduino IDE Library]. Retrieved July 12, 2022. https://github.com/claws/BH1750

Negroponte, N. The Architecture Machine, Cambridge, Mass., USA. The MIT Press, (-1970)

Painless Mesh Library. (Version 1.5.0) [Arduino IDE Library]. Retrieved July 12, 2022. https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home

Petersen, K., & Nagpal, R. (2017). Complex design by simple robots: A collective embodied intelligence approach to construction. Architectural Design, 87(4), 44-49.

Reynolds, C.W. (1987). Flocks, herds and schools: A distributed behavioral model. Proceedings of the 14th annual conference on Computer graphics and interactive techniques. (pp. 25-34).

Theraulaz G., Goss S., Gervet J., and Deneubourg, J.L. (1990) 'Swarm Intelligence in Wasps Colonies: a case for tasks assignment in multi-agents systems', presented at the 1990 IEEE International Symposium on Intelligent Control, Philadelphia, PA.

Valentini, G., Ferrante, E., & Dorigo, M. (2017). The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. Frontiers in Robotics and AI, 4, 9.