Master Thesis

Self-Guided Cooperative Transportation using a Swarm of Mobile Robots

(自己誘導するモバイルロボット群による協調輸送)

Ysmaldo Landaez

Advisor: Hitoshi Iba

January 28, 2019

The University of Tokyo

Department of Information and Communication Engineering

Graduate School of Information Science and Technology

Dedicated to my mother, Lisbeth Garcia,

to my father, Jose Landaez,

and to my brother, Efrain Landaez

ACKNOWLEDGEMENTS

My Biggest thanks to my family, which always support me in every step of my life and always have been giving me their blesses even though we find ourselves far apart. To my father, Jose Landaez, who always gave me the best words in my good and bad times, pushing me to move forward with a "Duro con ellos!!" at the end of each conversation. To my mother, Lisbeth Garcia, who gave me all the love and support when I felt down and, most important, when I felt happy. Always present with me as close as a click on the phone. Missing you since I left home. To my brother, Efrain Landaez, who taught to me how to be a loving brother and always reminding me how to be funny with innocence. This work is a message for you to never give up on your dreams and to know that with big efforts and perseverance always comes big results. Please continue as smart an as good hearted as always. To all of you, I will not have the time to thank everything you have give to me, but little by little I will try to do it. To my old friends that were in contact with me and year by year being present, always catching up to see how much we have grown in our life (Miguel Blanco, Josmel Suarez, Jofrank Gomez, Kevin Acosta). To my Venezuelan friends, Laura Rossetto and Angely Diaz, which have been present supporting me and cheering me with jokes or just by having our moments of being Venezuelans. Also helping to each other in difficult situations regarding the life as a foreigner. To the professor Iba Hitoshi for facilitating a space in his laboratory to develop this project. To Professor Hiroshi Dohi who gave me good insides after presenting my ideas. Always being helpful in order to mold an action plan. To my colleagues Nagar Anthel Venkatesh Suryanarayanan, Hiroshi Kera and Pindur Kotaro (Adam) from The University of Tokyo, Department of Information Science and Technology for helping me in the project with very interesting insides, discussion and the realization of this thesis. To my new friends whom I shared wonderful moments and adventures in Japan. To the people that appeared in my life, in order to give me help when I need it the most. To the Japanese government, which support us in order to pursue research in a Japanese University with an important scholarship Monbukagakusho. To the University of Tokyo for giving me the opportunity to belong to such an amazing institution. Of course, after I worked myself off to pass the entrance exam and pass the interview right after having 6 month of screenings to be granted with the scholarship. But thank you for letting me try and succeed.

ABSTRACT

SELF-GUIDED COOPERATIVE TRANSPORTATION USING A SWARM OF MOBILE ROBOTS (自己誘導するモバイルロボット群による協調輸送)

MARCH 2019

YSMALDO LANDAEZ

M.Sc., The University of Tokyo

Under the supervision of Professor IBA, Hitoshi

This study shows the benefits of self-organized guide robots to work with OBP in order to retrieve objects from any unknown environment, in a process we called Guide-Based OBP. The objective is to distribute the agents on these environments and complete tasks autonomously. The previous methods can accomplish this task only in open spaces or by manually guiding the robots to push the object. In this research is shown the implementation of the OBP method with different group of robots to determine its efficacy. These experiments are done by implementing plug-and-play robot team, where the agents cooperatively push an object towards the destination inside of an unknown environment. Subsequently, the Guide-Based OBP method is introduced, which consists on adding a guiding process to work together with the OBP method. The guiding algorithm establishes sub-goals between the object and the final destination, which helps the robots to push the object through the unknown environment. Based on the results of the simulations performed in V-REP. The results clearly demonstrate the effectiveness of the proposed method over existing methods in terms of speed and reliability.

Contents

List of Tables

List of Figures

Terminology of the Research

- Agent: In robotics, it has two meanings: (1) a robot or (2) a program that gathers information and accomplish some task without any supervision. For this project, an *agent* is a robot.

- Swarm (Multi-robots systems): Consist in a large amount of simple robots. The aim of having such amount of robots is to produce cooperative behaviours, found in nature, from the interaction of the robots and their environment.

- Environment: In simulated and physical robotics, is the compound of circumstances, obstacles, objects and other agents by which a robot or a group of them are surrounded.

- Open Environment: It is an environment without obstacles, where a robot can see all the targets within a 360 degrees turn.

- Unknown Environment: This term applies to the perspective of the swarm of robots, given that none of them as individuals or as a group have any information about the environment; such as maps, coordinates or cameras above the environment. Moreover, the robots do not form a map during the task (no SLAM, no mapping algorithms applied), which means that in any moment the robots know where they are.

- Single Track (Unicursal) Environment: are those which the inner path can be traced with one uninterrupted line passing through all the free spaces. Also called, non-branching or "unicursal" [Wik18] environments.

- Maze: is a path or collection of paths, typically from an entrance to a goal [Wik18].
- Maze-type Environment: are those which the inner path can be branched in several ways. In other words, it is an environment that can have different routes from the starting point that might lead or not to the goal. In the worst case those paths also might not lead to the returning point.
- Image Processing: It refers to any process involving mathematical operations or algorithms to extract features or enhance and image.
- Occlusion: The complete obstruction of any physical need or activity; such as breathing, observing, hearing, by another object or person.
- Occlusion-Based Pushing (OBP): Is an algorithm proposed in 2015 by Jianing Chen, in which explains how a robot can occlude its vision of the Goal it has to reach by the object it has to push. By this method a robot can retrieve any object in an open environment.
- State Diagram: is a type of diagram used in computer science and related fields to describe the behavior of systems. The system described is composed of a finite number of states.
- State: In information technology and computer science, a program is described as stateful if it is designed to remember preceding events or user interactions; the remembered information is called the state of the system.
- Plug-and-Play: From *techterms.com*, Plug and Play, sometimes, abbreviated PnP, is a catchy phrase used to describe devices that work with a computer system as soon as they are connected.
- Goal (OBP): Is the target where the robots have to push the object to rescue or retrieve. Furthermore, it is the closest, visible target to reach.
- Home (Guided Pushing): Is the final target place where to push the object to retrieve. It is called home because, it is the place where the robots are deployed

and where they have to go back after finishing the task; Such as the nest of an ant colony.

- Pheromone Trail: are semiochemicals secreted from the body of an individual to impact the behavior of another individual receiving it [Wik17]. Many species of ants leave a trail of pheromones when carrying food to the nest. Ants follow the trails left by other ants when searching for food [Iba13].

Chapter 1

Introduction

1.1 Motivation

The need of exploring unknown or hazardous zones, such as collapsed buildings, without risking lives opened a research and development field for roboticists in order to create agents able to explore and extract information safely. Nowadays, the robots created for this aim are generally large or with human shape not making them able to reach narrow and intricate spaces. As a result, smaller robots are preferred. Especially, when the transportation of an object inside these areas is required.

The main source of inspiration for swarm roboticists comes from observing the behavior of social animals and, in particular, social insects such as bees, ants and termites. Studies have revealed that there exists no centralized coordination mechanisms behind the synchronized operations in social insects, and yet their system's performance is often robust, scalable and flexible. These properties are desirable for multi-robot systems and can be regarded as motivations for the swarm robotic approach [Bon+99].

The robot systems can be separated into two categories: centralized and decentralized systems. In centralized systems, a single camera observes the environment and the robots as a whole, and a central computer processes the camera information to remotely control each robot in order to accomplish a task. Another important characteristic of these systems is that they possess a leader, which guide the rest of the robots. These facts, lead to several limitations that decentralized scheme overcome. For instance, the leader's death means the loss of the team's guidance.

The transportation of significantly large objects by robots have been studied since the industry started to develop. Engineers [Ing+17; Bas+08] and researchers [Yam+00; Alk+15; Bas+08; SIC; Mec05; Tor15; Wan+05], have done plenty of work on this topic. For tasks such as [Yam+00], two large robots were used; but these solutions tend to become expensive for researchers and companies. Nowadays, we can obtain computationally powerful robots at an affordable price with good hardware quality in order to implement our research [EPF; Cor]. But for transportation, conventional robot hardware does not offer the power and torque necessary to move large objects. Nevertheless, researchers have developed the idea of using swarm intelligence on low cost-robots in order to overcome these limitations.

Why using a swarm of robots? Although each of the agents is very simple, they can carry out complex tasks when they work together; emerging what is called Swarm Intelligence. The robots can work in parallel, so one group of agents can be performing a task while the rest can perform another task entirely. Swarms of robots are Robust to failures, if one of the robots fails, other robots can keep carrying out the task. This type of swarms are categorized as decentralized.

Our method is based on simple instructions that the robots must follow depending on what they observe in the environment. A simulation is an obvious choice to quickly evaluate the performance of the group of robots. Therefore, all our experiments were implemented in simulations. The majority of applications using simulators to produce a decentralized swarm of robots apply evolutionary computation approaches. These methods are time consuming due to training processes and the robots must learn within the same environments, we found this unpractical for our aim.

The aim of this thesis is to create an algorithm that, applied to each member of a decentralized swarm of robots, can be deployed into an unknown or hazardous area. The proposed robots are small with the purpose of make them able to enter in narrow spaces to explore and retrieve objects, as in a real rescue application.

1.2 Research Goal

We aim to create a swarm of robots that can be deployed to enter in narrow spaces and retrieve an object by pushing it. This pushing process would be guided by the same robots involved in the retrieval task. Furthermore, this swarm system is designed to work as a plug-and-play equipment and use low cost hardware as well as software. Knowing that a single robot with this characteristics, cannot accomplish some specific tasks by itself, we rely on the robustness of the cooperative behavior of a group of robots programmed with simple instructions in order to retrieve an object from any unknown environment.

1.3 Challenges and Contributions

In this thesis we address a new problem, which in the previous work regarding OBP [Che+15] solved it in a manual way. Our approach is to use a swarm of mobile robots in order to retrieve objects, where the robots do not communicate between each order or to a central computer. These make our swarm of robots a decentralized system.

The decentralized systems operate without a leader, leaving the decision making and the completion of certain task to each robot as individual. Given this, the robots should have enough sensors (such as: cameras, distance sensors, etc.) in order to obtain sufficient information from its surroundings and produce a similar behavior as found in nature [Tor15]. The advantages of these systems are robustness, scalability, flexibility and collective behavior.

The OBP only works in open spaces, where the robots can observe the goal and the object after a 360 turn on their spot. If there are obstacles between the object and the goal then, it is required to manually put guide robots or remotely control one of them to be followed by the rest. These obstacles can be represented as walls in a room.

We propose a method that guides the group of robots using OBP through unknown environments with obstacles not requiring human manipulation nor remote controlled robots. Our algorithm allows the robots to produce a path, without communication between each other, in order to lead the pusher robots where to bring the object back to the destination. This path will work similar to the pheromone trail of an ant community. The robots that take up the role of creating a path will be called guides. In the later sections, a more comprehensive analysis of the algorithm will be presented.

1.4 Thesis Outlines

The thesis is organized as follows:

The Chapter 2, presents the related work of object transportation and cooperative transportation. Then, it studies the limitations of using evolutionary computations to solve the search and retrieve application in rescue situations. Finally, this chapter concludes by explaining the method of transportation used to retrieve an object (Occlusion-Based Pushing).

Chapter 3, describes the problem this thesis asses and highlights the limitations of the Occlusion-Based method in order to retrieve an object from an environment with obstacles. After analyzing the vulnerabilities of the previous method, it is presented an algorithm to overcome this weakness and work alongside with the OBP called self-Guided OBP.

Chapter 4, Introduces the robot and its sensing devises, Khepera III, used for the following experiments within the designed environments produced in VREP 3D simulator. Moreover, it describes the functions of the robots and its configuration in order to explore the environment and accomplish the proposed tasks.

Chapter 5, describes the experimental set-up and presents the results of the experiment done within the environments created in VREP. Moreover, it offers snapshots of the videos taken of the simulation in order to highlight the novelty and contributions of the self-Guided OBP.

Chapter 6, offers a detailed explanation of the expectations of this research and how many of them were covered. Additionally, the results and limitations of the proposed method are discussed regarding what it have been achieved.

Chapter 7, concludes the thesis, discussing the benefits of the self-Guided OBP in contrast with the traditional OBP method used for cooperative object transportation. Finally, the future work for this research is assessed by giving new applications in other fields and directions to modify the proposed algorithms in order to improve its performance.

Chapter 2

Previous Works on Swarm Transportation

The object transportation task using autonomous systems has been approached and applied in many works. The objectives and robot architectures used are very different as well as the algorithms they implement in the robots. A collective transport strategy, inspired by the food retrieval procedure of ant colonies (Figure 2.1), has been implemented on a swarm of robots that are smaller than the object. The three most common types of strategies are pulling, pushing, and caging [Che+15].

Beginning with the opposite but important task to study, the work in [Mel+15] considers the transportation of multiple objects dispersed in the environment using the non-prehensile manipulation method with a single robot making emphasis on the path planning and task allocation phase. A push-based manipulation of objects is used here similar to the proposed algorithm for object manipulation. The objective is to minimize the time to get all the objects the their specific goal positions also minimizing the distance traveled by the robot. The problem is modeled as a graph problem, where a path is decomposed into segments and the robot dynamically selects in a greedy manner the best to be executed in a certain time. The tasks were simulation-based considering a real robot in order to provide evaluation and validation of the methodology. This research gives an idea of what a single robot is capable to do in a transportation application.

Figure 2.1: Example of ant colony cooperative transportation. Roboticists always have found inspiration to build robots or emulating behaviors they observe in the nature. Therefore, the cooperative transportation field, seeks to use robots to imitate the ants behavior at the moment of transporting food from the source to the nest.

Nevertheless, this task could be solved faster if a group of robots are implementing to accomplish the same objective. Furthermore, if the implementation of a robot is executed in a simulator, this one should be capable of recreate the noise and real characteristics of the environment and the agent, in order to be more reliable at the time to be tested and used in a real application.

Using multi-agent for the task of object transportation with real robots, it is found the transportation of a fixed object through an environment as a way of introducing a multi-robot system-based optimization [Sad+16]. Two robots jointly carry a stick from an assigned initial position to a specified final position in a given environment, without collision with the given obstacles near the robots and the stick. The sensory data of the robots are the input variables of the optimization problem while the output variables are the necessary amount of rotation and translation of the stick (by the robots) to transfer it in small steps towards the goal. The evolutionary optimization approach of solving a multi-robot stick-carrying problem proposes a novel strategy to embed the motion dynamics of fireflies of the Firefly Algorithm (FA) into a socio-political evolution-based meta-heuristic search algorithm, known as the Imperialist Competitive Algorithm (ICA). This hibridization is called Imperialist Competitive Firefly Algorithm(ICFA). This application justifies the importance of the proposed hybridization and parameter adaptation strategies in practical systems. This application is based on the ideal that the object to be transported is already attached to the agents. Therefore, the autonomy of the robots gets limited just for the retrieval case of a real search and rescue situation.

The research on [Alk+15], presents a set of simulations in which autonomous robots are required to coordinate their actions in order to transport a cuboid object that is too heavy to be moved by a single robot. They compare two different hardware: (1) NT-condition, robots equipped with a camera and proximity sensors and (2) T-condition, are robots that possess torque sensors additionally. The result shows that best evolved groups of the T-condition outperform those of the NT-condition. Moreover, we show that the best evolved groups can adapt to variability in size and weight of the object as well as to the small variability in the cardinality of the group. We also show that simple forms of recruitment behaviour emerges without being selected for during evolution. This work unveils interesting relationships between design choices and characteristics of the evolved solutions, and it contributes to develop design guidelines for engineering robust and successful swarm robotic systems.

In [Hab+18] presents two distributed algorithms for enabling a swarm of robots with local sensing and local coordinates to estimate the dimensions and orientation of an unknown complex polygonal object, i.e., its minimum and maximum width and its main axis. Their first approach is based on a robust heuristic of Distributed Principal Component Analysis (DPCA), while the second is based on turning the idea of Rotating Calipers into a distributed algorithm (DRC). They simulate DRC and DPCA methods and test DPCA on real robots. The result shows the success of the algorithms to estimate the dimension and orientation of convex or concave objects with a reasonable error in the presence of noisy data. This work can be useful for transportation methods given that it is possible to calculate the amount of robots needed to move an object knowing its estimated dimensions.

The research done by [Bas+08] shows that an appropriate human interaction can benefit a swarm of robots to achieve goals more efficiently. A set of desirable features for human swarm interaction is identified based on the principles of swarm robotics. They created an indoor environment with the help of a simulation program using MATLAB. The swarm behavior and the results of user interaction are studied by considering radiation source as the goal for a search and localization application with the swarm. Particle swarm optimization algorithm is slightly modified to enable the swarm to autonomously explore the indoor environment for radiation source search and localization. The emergence of intelligence is observed that enables the swarm to locate the radiation source completely on its own. Proposed human swarm interaction is then integrated in a simulation environment and user evaluation experiments are conducted. Participants are introduced to the interaction tool and asked to deploy the swarm to complete the missions. The performance comparison of the user guided swarm to that of the autonomous swarm shows that the interaction interface is fairly easy to learn and that user guided swarm is more efficient in achieving the goals. The results clearly indicate that the proposed interaction helped the swarm achieve emergence. A common feature with this study is the usage of an object that robots can recognize as the target to be transported, as well as the search and localize application. The limitation that our research aims to improve is the elimination of the human interaction stage, which servers as a guide to the robots that possess the object ready to retrieve. Therefore, applying an autonomous search, guide and retrieve method with the same group of robots will lead to a better retrieval process using an optimal number of robots.

In [Che+15] the proposed strategy for transporting a large object to a goal is to use a large number of mobile robots that are significantly smaller than the object. The robots only push the object at positions where the direct line of sight to the goal is occluded by the object, this process is called Occlusion-Based Pushing (OBP). These experiments were implemented in an open environment which makes easy to the robots to glance both targets (the goal and the object) in a single turn. In other experiments, using complex environments, they apply guiding methods in order to transport the object back to the goal position. 1) Using a group of robots placed by hand, lighting the color of the goal (red) and 2) implementing a leader guide robot manipulated remotely. As mentioned in the last paragraph, this guiding process, which involves human intervention, it is aim to be absorbed by the swarm.

2.1 Limitations of Evolutionary Computation

The traditional evolutionary method used to achieve similar tasks of rescue or object retrieval requires long-time training and computationally complicated algorithms, as briefly described in the last section, which involves the following undesirable factors for our project:

- Same environment: The usage of the same environment is needed for the learning process because the agents should reset the its positions after an epoch or episode is finished. By doing this, we can ensure that the robots will optimize the task policy faster because they wouldn't need to use other algorithms to self-locate in the environment spending computational resources and time.

- Large number of sensors: Other EC algorithms works better when each agent has a large number of sensors, in order to take better decisions to improve the policy, such is the case of Neural Networks (NN) [Isl+17] or Optimization algorithms. This solution is computationally expensive (if the experiments are run by simulations) and economically (due to production and design costs)

- Image Processing: As well as the previous point, image processing requires large computational resources due to the input, that instead of being a large number of sensors, is a large number of pixels with color information. After these data are read, the algorithm is expected to classify these pixels to determine which group of actions are suitable to accomplish the task. The classification and decision making is achieved by reaching the highest reward possible.

2.2 Occlusion-Based Pushing (OBP) as Transportation Method for Swarm

This section briefly describes the method we base our research in order to achieve the goals explained in Section 1.2. Occlusion-based pushing [Che+15] consists of pushing an object as long as the robot cannot observe the goal (or the source of light or color). The goal is the immediate place where the robots have to push the object. Even though the underlined specification of this method is an important limitation in real-life applications, it has inspired our project in order to use it as a transportation framework. Given that,

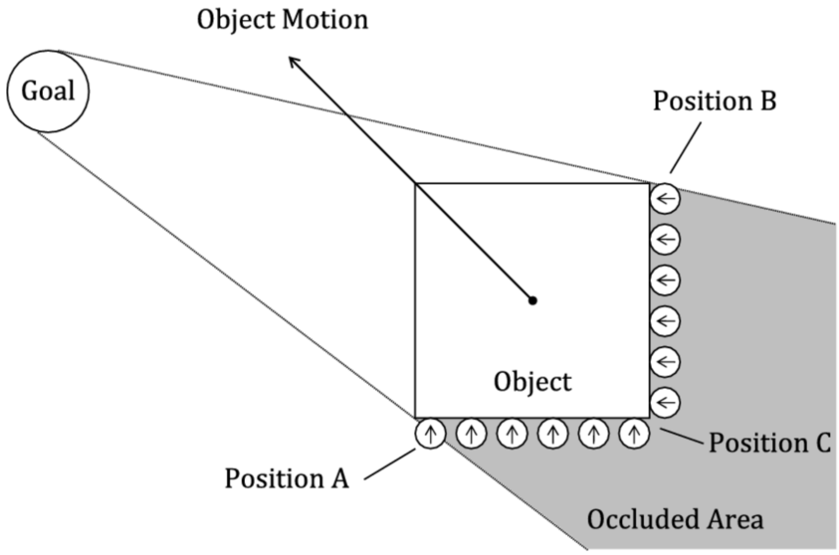


Figure 2.2: Illustration of how a swarm of robots can push a large object in a 2-D planar environment. As observed in the image, the robots push the surface of object inside of the occluded area. This process results in a motion that not necessarily lead the object directly to the source. However, the robots are able to correct their positions if the conditions of the OBP are not fulfilled in a point of time. Figure taken from [Che+15], page 3, Fig. 1.

neither the robot or the object has no specific way to get attached to each other; pushing is the best way to move it by the force of one or more robots. Until now, Chen [Che+15] has given the most efficient method, considering the latest researches in the area.

A positive aspect of this transportation strategy is that, rather than treating occlusion as a problem to be overcome, occlusion is used to organize a swarm of robots to push a large object to a goal. The algorithm deals with pushing the object across the portion of its surface, where it occludes the direct line of sight to the goal. This results in the transportation of the object along a path that may not be optimal, but always

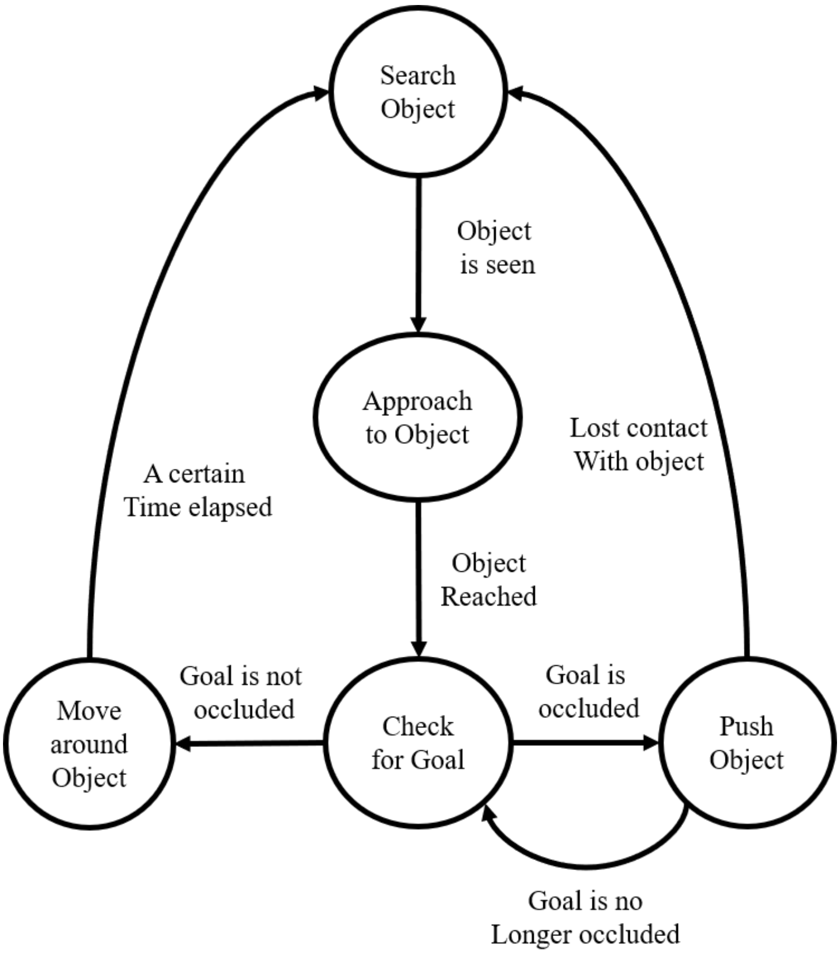


Figure 2.3: State machine representation of the individual robot behavior realizing the occlusion-based cooperative transport strategy. Figure taken from [Che+15], page 3, Fig. 2.

arrives at the goal. In Figure 2.2, we highlight the presented theory of occlusion.

The simplicity of the strategy makes it particularly suited for implementation on mobile robots that have limited capabilities. In the long term, such simple multi-robot strategies could be implemented at very small scales and for important tasks such as rescue scenarios.

Figure 2.3 presents a State Diagram Representation of the algorithm presented by Chen. This state machine representation produces the following behavior: once the object is seen the robot moves toward it ("Approach Object"). When the robot has reached the object, it enters to state "Check for Goal" to realize whether the goal can be seen from the robot's position. If the goal cannot be seen, the robot will push the object simply by moving against it ("Push Object").

As briefly explained in the previous section, Chen [Che+15] use this strategy in order to transport and object back to a target point called Goal. In order to accomplish this task, they use e-puck robots due to its simplicity and affordable hardware in order to produce the swarm of robots.

They made several testings with different shaped objects as well as different object masses, in order to validate the proposed algorithm. Nonetheless, we are going to focus in the experiments and results they have done with a circular object. The results shown by [Che+15] can be observed in Figure 2.5.

In the same research, they perform a second experiment using the same circular object and arena as before, but two walls were added to serve as obstacles. The destination was a rectangular region opposite the initial position of the object. The direct line of sight between the object's start position and the destination were blocked by the walls (see Figure 2.6).

They realize this experiment using the environment area shown in Figure 2.6 and using the same algorithm to program the robots. In other words, these e-pucks are programmed to push a blue object to a red goal. Nonetheless, the robots wouldn't know were the object or the goal where by themselves. In order to overcome this limitation, they used another e-puck vested with red with the objective to guide the pusher robots

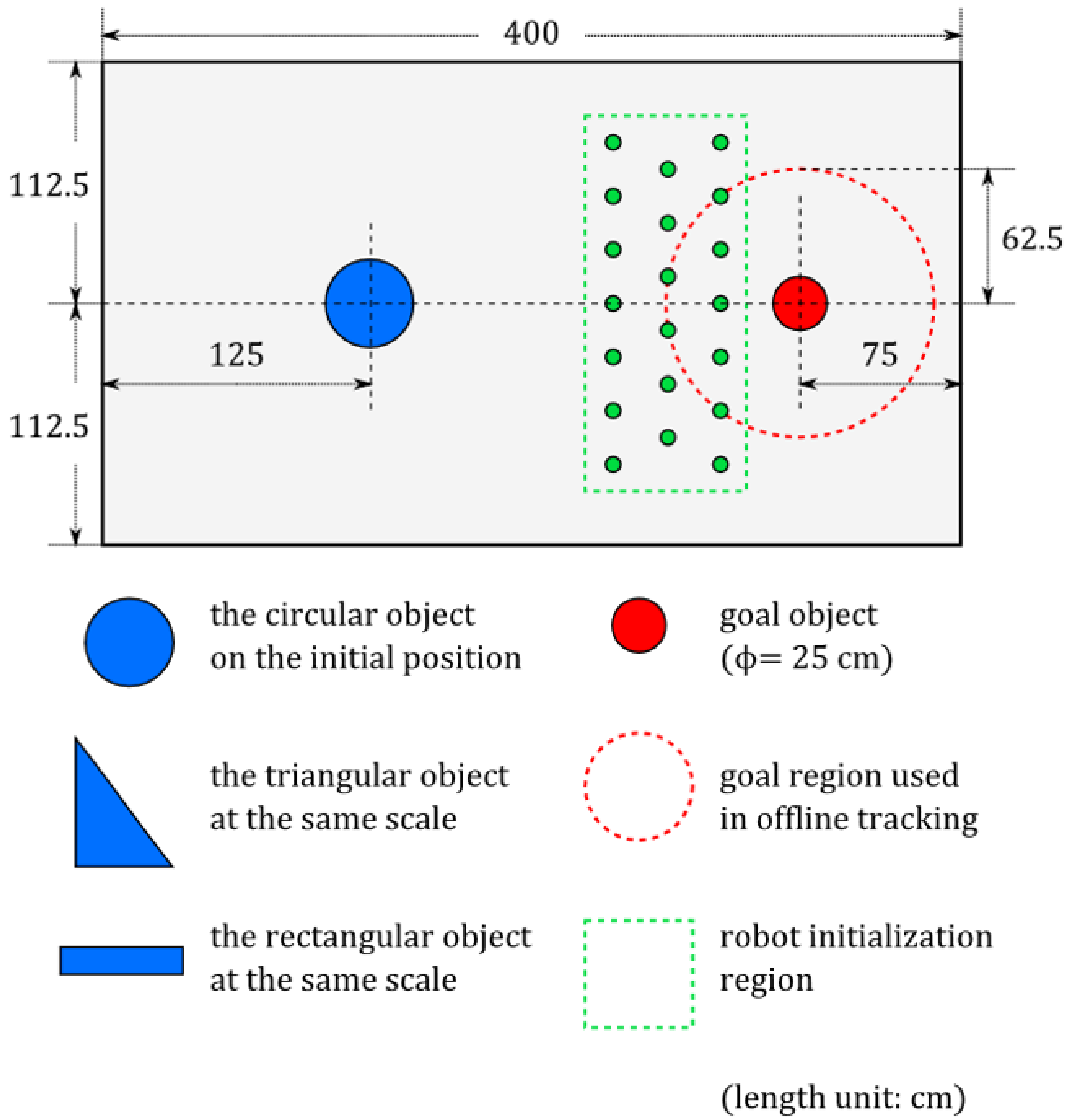


Figure 2.4: Experimental set-up. Notice that the second and third blue shapes (the triangular an rectangular object) on the bottom left of the image are experiments that will not be taken into account. Figure taken from [Che+15], page 7, Fig. 5.

to the Goal position as a mobile goal (the goal robot). To further increase its visibility, it kept all of its red LEDs turned on. The goal robot was programmed to be driven

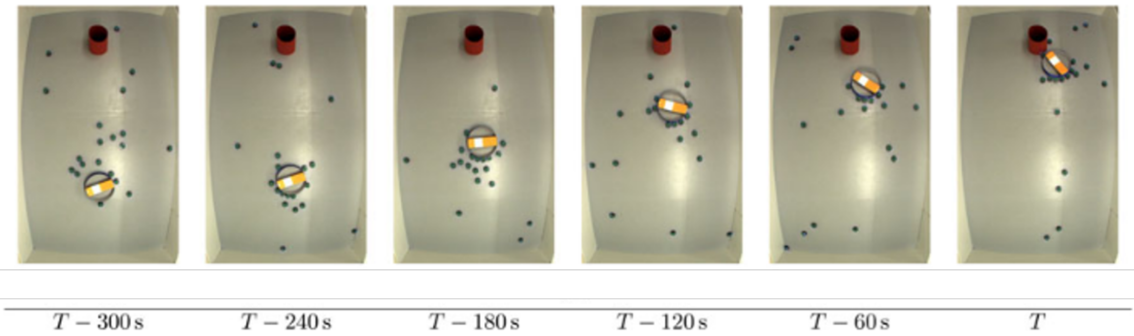


Figure 2.5: Experiments with real robots (e-puck), pushing cooperatively a circular object. Figure modified and obtained from [5], page 8, Fig. 7. The "T" letter represents the total duration of the videos (in seconds). Following the time format of this thesis T = 5minutes. This time is just measuring the duration of the pushing process.

remotely by a human operator via Bluetooth. As the transport robots push the object toward the goal robot, the operator can indirectly control the transport direction by driving the goal robot. The experiment results can be observed in Figure 2.7.

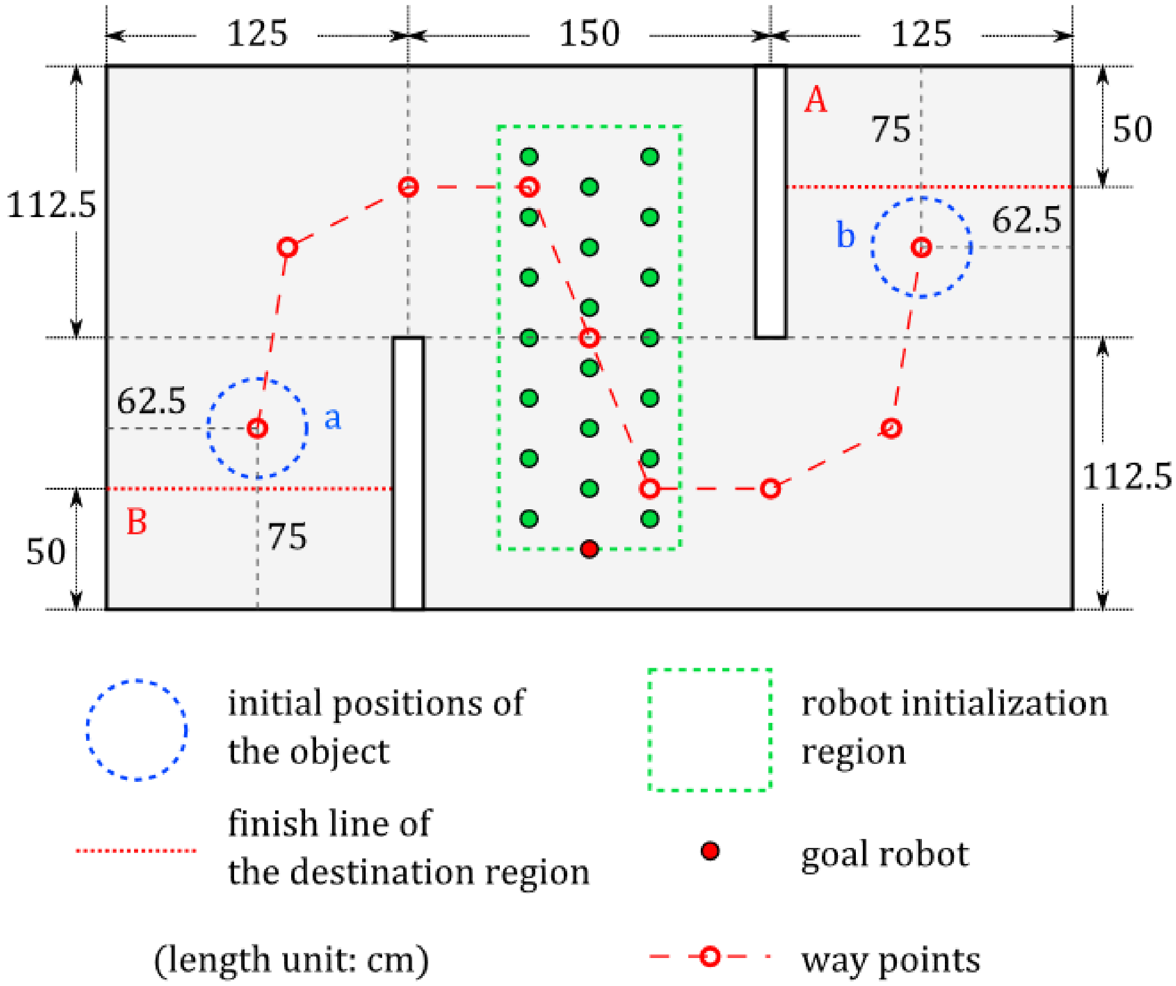


Figure 2.6: Experimental set-up with a moving goal. The goal robot is represented by a red dot. The completion of the object retrieval in this environment, is ensured due to the movement of the goal robot through the way points until the Finish line. Figure taken from [Che+15], page 11, Fig. 12.

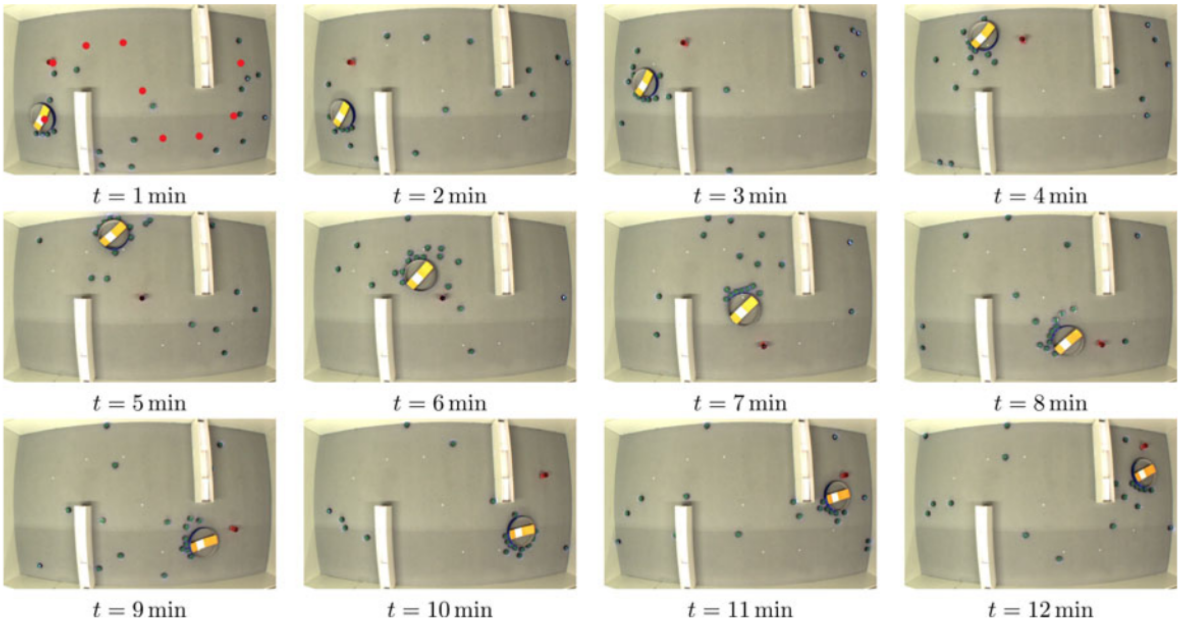


Figure 2.7: Snapshots of the experiment where the transportation group pushes the object toward a teleoperated goal robot through an environment with obstacles. In the first snapshot (t = 1 min), the way points for the goal robot are highlighted. Figure taken from [Che+15], page 12, Fig. 13.

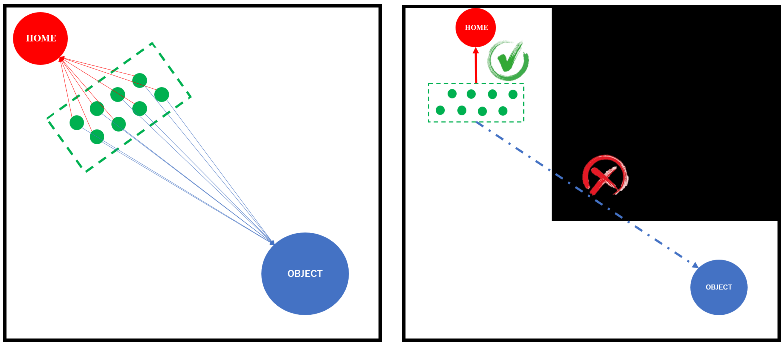
Chapter 3

Self-Guided Cooperative Transportation Method for a Decentralized Swarm of Robots

3.1 Problem Description and Conditions

The task that we considered is as follows. A bounded environment contains a cylindrical object, a Home position, and a number of robots. The environment contains wall that serve as obstacles. The aim is that the robots, which are initially placed close to the Home location, push the object to the goal through the environment sorting the obstacles. Note that the Home position specified in the problem is aimed to be the final destination of the transportation. In all the scenarios proposed here, the Home location would be static, encouraging the robots to guide the pusher team if the direct sight to Home is blocked by the obstacles in the environment.

We make the following assumptions. The object and Home targets can be recognized by the robots. The robots can not communicate between each other, thus they must interact with the environment to find the correct solution for their actual status in its surroundings. The dimension of the object is large enough to occlude the robots' perception of the goal when they are behind it [Che+15]. The Home is placed close from



(a) (b)

Figure 3.1: (a) Open space environment for OBP. Lines showing that every robot can see the goal and the object. Non-open environment (Experiment 1). (b) Visualization lines showing the limitations of the OBP method in environment a non-open environment.

where the robots will be deployed. Nonetheless, the robots do not have information about the Home location. If the robots can perceive Home and the object from its initial position they will perform the traditional OBP method. In the case that they just perceive Home, it will be assumed that the object is in another location of the environment. Therefore, the robots will perform the proposed Guide-Based OBP method introduced in section 3.3.

3.2 Previous work limitation

As we briefly explained in the Chapter 2, specifically in section 2.2, the limitation of the occlusion-based pushing method is, the need of an open space to produce the expected results. In this way, the robots can see after a complete turn the object and the goal. Refer to Figure 3.1a.

In their work [Che+15], they present two ways of pushing an object to the destination through an environment, where it is not possible to have a direct glance of the object nor the goal in a 360-degree turn. The first idea is to place, manually, robots forming a path with sub-goals towards the destination; then, when the object is close to the placed guide, this one will start to help the other robots to push.

The second approach is to cover one robot with the goal's color as a guide. Then, with a remote-control, move it through the environment until it's close to the actual destination, while the rest of the robots are pushing the object.

Knowing the limitations of the traditional method (Figure 3.1b), as well as the manual solutions the researchers implemented, we propose an algorithm that allows the swarm to search and push an object through a non-opened environment. However, The conditions of the robots in order to retrieve the object are strict:

- The robots (nor individually, not shared by the group of robots) do not have any information about the space surrounding them. In other words, there is no input to the robots that tells them where they are, nor algorithm that produces a path or a map, such as:
  - Ceiling camera. In fact, this method is common in centralized systems which is not our case.
  - SLAM (Simultaneous Localization and Mapping) or Mapping algorithm.
- The robots do not have a way to communicate between each other (such as, Bluetooth, WiFi or RF (radio communication) signals) in order to send their position or any instruction regarding the next step to take.

Regarding the conditions described above, we have designed an algorithm that overcome this limitations by relying on the cooperative behaviors the swarm of robots.

3.3 Self-Guided OBP for Cooperative Transportation with a Swarm of Robots

In our framework, each robot is able to perform one of this two states interchangeably:

- Push Mode: a robot pushing the object by applying the OBP method (Figure 2.3).

- Guide Mode: a robot that place itself in a strategic position as a sub-goal to lead the robots in the Push Mode (Figure 3.3).

Therefore, when the conditions for the traditional OBP to work are not fulfilled (Section 2.2), our algorithm allows the robots to explore the area in order to decide whether to take a guide role or a pusher role by themselves (Figure 3.2). The traditional OBP is a simple but powerful algorithm that makes a single or a group of robots to push an object towards a target called goal. But the algorithm itself needs special and no realistic conditions, when it is applied to any environment with obstacles. As we mentioned in the previous section, the obstacles block the line of sight between the robots and the targets they should perceive (the object and the goal position). Therefore, guiding robots are necessary to let the OBP method to retrieve the object to home successfully. Moreover, human intervention is not an advantage when the environments where the robots are deployed is unknown [Bas+08]. Consequently, the guiding robots should work autonomously as the traditional OBP.

The Home is represented by the red color in Figure 3.1. The difference between "Goal" and "Home" are: The former is used in the OBP method developed by Chen [Che+15], because it is the immediate place where the robots have to push the object. The latter refers to the final destination of the object, which is the place where the robots are deployed. In our method, the "Goal" of the traditional OBP algorithm (applied along with the Guide Mode) would be the sub-goals represented by the guide robots.

The proposed self-Guided OBP algorithm is conformed by two states. These are governed by the decisive factor of the first state called "Check Home and Object". The aim of this initial state is to determine if home and the object are visible within one 360 degrees turn. If the both targets are observable in the first check, the robot will perform the Push Mode, (refer to Figure 3.2, left state) which is the traditional OBP algorithm. In the case that home is the only target perceived, the robot will perform the guide mode algorithm (refer to Figure 3.2, right state).

The Guide Mode is composed by 7 states. However, the "Explore for" and "Approach to" states are repeated, which means that they follow the same working principle but

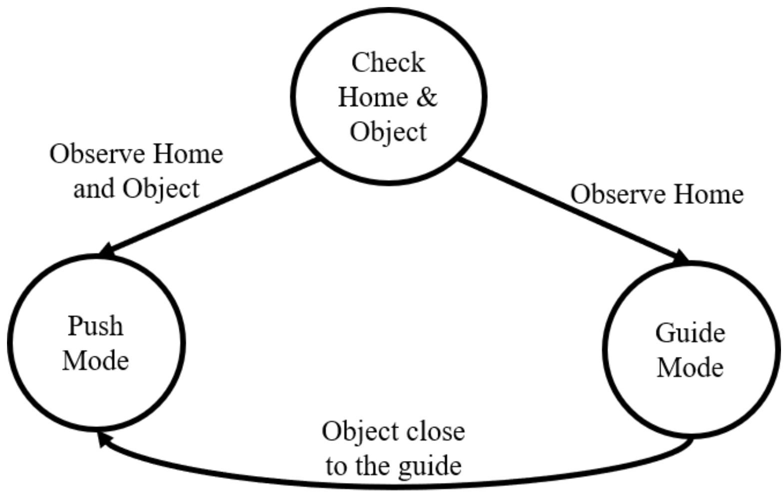


Figure 3.2: State machine of the first stage of the proposed algorithm. On the left side, under the name of “Push Mode”, the traditional OBP method is implemented as an independent function. On the right side, under the name of “Guide Mode”, the proposed self-Guided method is put into effect also as an independent function working alongside with the Push Mode.

with different targets. The first state is called “Explore for”. This state is performed by a Braitenberg [4.2.1] obstacle avoidance method [Sha+12], in order to evade the walls that serve as obstacles within the designed environments and robots around. The exploration will be performed until the agent can detect one of the targets (Home, a guide or the object).

The next states, “Approach to” and “Check Home & Object”, are implementations from the traditional OBP used in the Guide Mode to accomplish the same purpose. In the right side of Figure 3.3, we can observe a special condition within the state “Approach to (Home or Guide)” that the agents have to satisfy in order to become a suitable guide. This state is going to be performed if there is no other robot in the line of sight of a potential guide robot. Otherwise, the agent would stop being proceeding to be a guide and will look for the object, in order to start the Guide Mode process again. This would ensure that two or more robots are not going to accumulate as guides in the same or a close position. The explained above is displayed with an example taken from

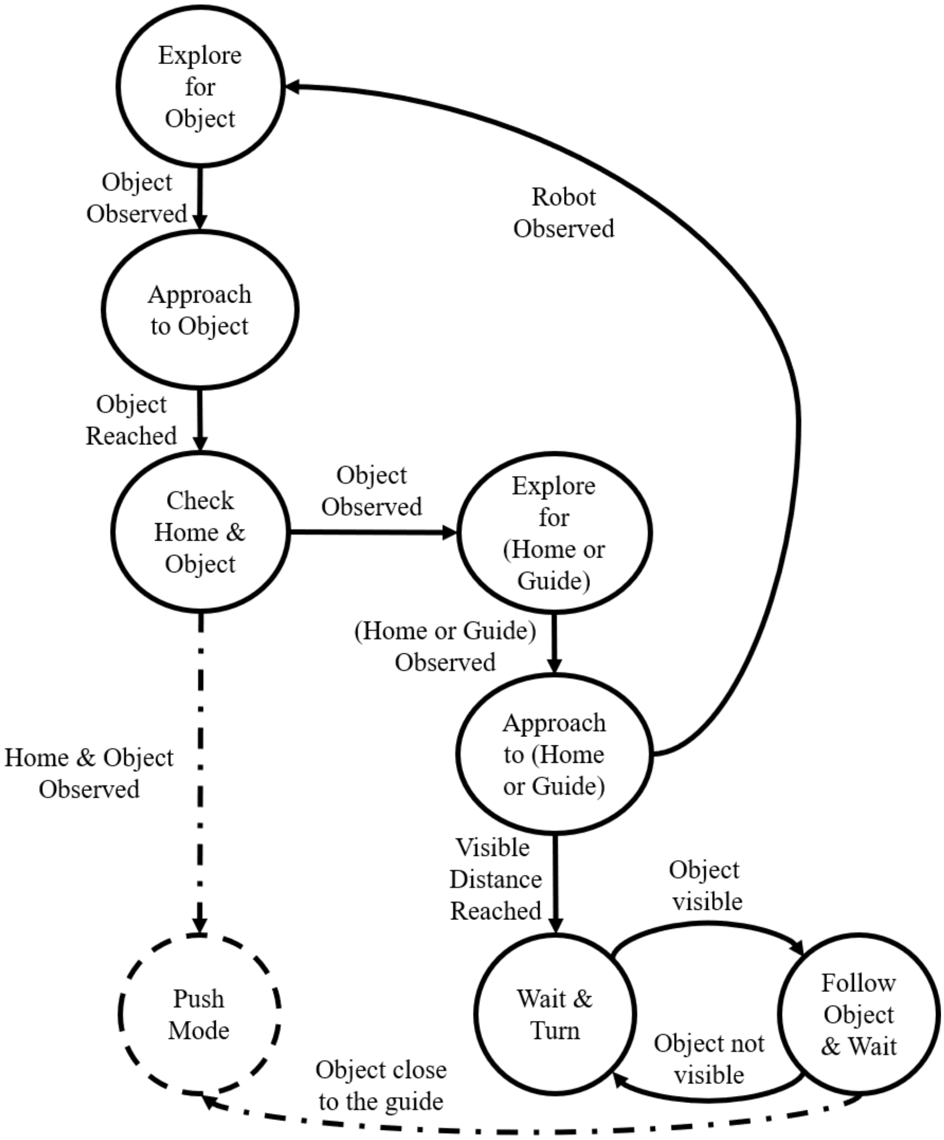


Figure 3.3: State diagram of the novel Guide Mode method. On the left side, the method used to find the object in the environment by the robot is introduced. On the right side, it is shown the states needed to determine how the robot can become a guide.

the experiment 2.b (Figure 3.4).

The last two states are the key of our proposed algorithm: “Light Color & Turn” and “Follow Object & Wait”. The former, refers to the change of color on the robot’s external case from green to orange (color used to identify the guide robots). The color

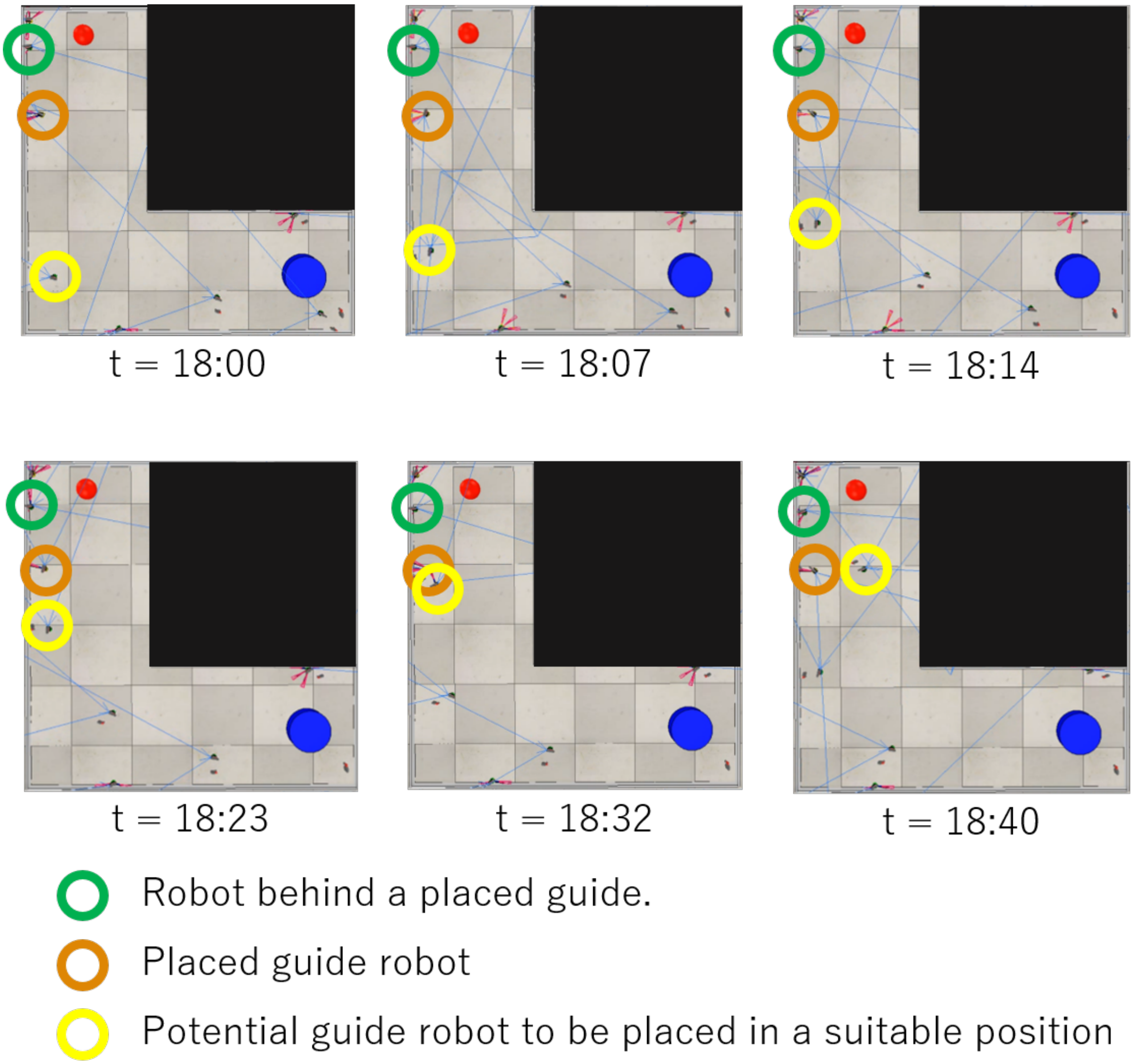


Figure 3.4: Condition of a guide robot to abandon its task. This snapshots were taken from the simulation on experiment 2.b (section 5.2.2, Figure 5.4. The fist image shows a robot with the purpose to become a guide a the minute 18. After 7 seconds, the robot starts to approach to the previously placed guide robot (in the orange circle), in order to estimate the necessary distance to be placed as a new guide. The third snapshot (t = 18:14), shows the moment when the potential guide robot's camera detects the robot behind the previous guide. The last three snapshots shows the behavior of a potential guide robot abandoning its previous duty.

changes after the robot reaches a distance were it can perceive the previous guide and other robots can perceive its presence. Then, the robot rotates on its own spot until the object is visible.

The latter is performed after the object is visible by the guide robot. In this state the robot perform small turns while observing the object, allowing it to keep the object perpendicular to its vision line. This behavior permit the guide robot to measure the shortest distance to the object from itself. Then, after the object is close enough from the guide robot, this would perform the traditional OBP method in order to help the team to accomplish the final task as well.

The algorithm (Figure 3.3) is read in the following way: the robot first explores the environment until it can observe the object. Then, the robot approaches to the object until it is reached. Here, the robot checks if Home or a guide is observable from its position. If it does, the robot start to perform the Push Mode towards the observed sub-goal. Otherwise, the robot explores back the environment until it can observe Home or the closest guide from it. If the robot observers the guide or Home and another robot, it will go back to search for the object (refer to Figure 3.4). If it just observes another guide or Home, it will place itself at a specific distance to be observed by the other robots and contribute to form the path.

When the robot is already placed in position, it starts to turn on its own center waiting for the first glance of the object approaching. If the object is perceived by the guide robot, it would stop turning and would follow the direction of the object, moving left or right if necessary, in order to keep it perpendicularly to its line of sight. When the object is close to the guide, this would change its state to the Push Mode.

Finally, when our method (self-Guided OBP) is applied to each robot, in conjunction, they are able to produce a path in the environment and therefore, let the rest of the robots to move the object using this guidance as a pheromone trail, such as the one ants use in order to know the way back to nest, until they are able to find home (refer to the result in the section 5.2.2 and 5.2.3).

Chapter 4

Selected Robot and Simulation Environment

In order to demonstrate how the proposed algorithm works, we have decided to utilize a simulator. Due to limitations in the hardware, we determined to use V-REP [Rob] as our rendering tool for the following experiments 5.

4.1 Khepera III as a Low-Cost Solution for Swarm

The chosen mobile robot is the Khepera III [Cor]. The device can be found in the simulator as a pre-made sample inside the robots' folder. The Khepera model is ready to used after is located in a place within the environment. After several tests to study its functionality, we realized that the mobile robot has embedded many features that are not going to be useful for the experiments. Additionally, if those features are kept, the simulations would last longer, due to the calculations of speed and position that the simulator must do in order to render the environment. Therefore, those features were reduced. This reduction also had the intention of saving computational resources. We can see the changes made on the robot in Figure 4.1.

The hardware of this mobile robot is large compared to its predecessors, but it possess a powerful CPU and more sensors available, which we are going to detail in the next image (Figure 4.2).

- 9 Infra-red proximity and ambient light sensors
- 2 Infra-red ground proximity sensors for line following applications
- 5 Ultrasonic sensors

4.2 Braitenberg Obstacle Avoidance Method

In this section we briefly introduce the topic of Braitenberg vehicles. These vehicles, named after their inventor neuroscientist Valentino Braitenberg, have been extended to address the design of autonomous agents in general [Pfe+01]. The basic theory of Braitenberg vehicles will be introduced in this section along with their examples. Furthermore, the application and modelling of the Braitenberg method in the Khepera

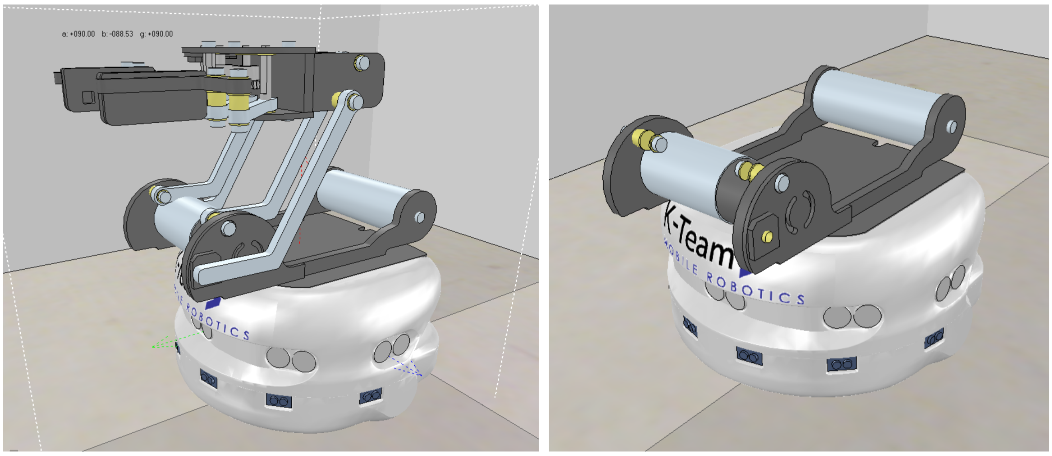


Figure 4.1: Changes in the robot. (a) shows the Khepera robot [Cor] that VREP [Rob] simulator brings as pre-made robot sample. The features to be reduced are: the mechanic hand or gripper which is embedded with a set of sensors, the gripper's camera and the motors in charge of moving this device. (b) Shows the mobile robot after the feature reduction explained previously.

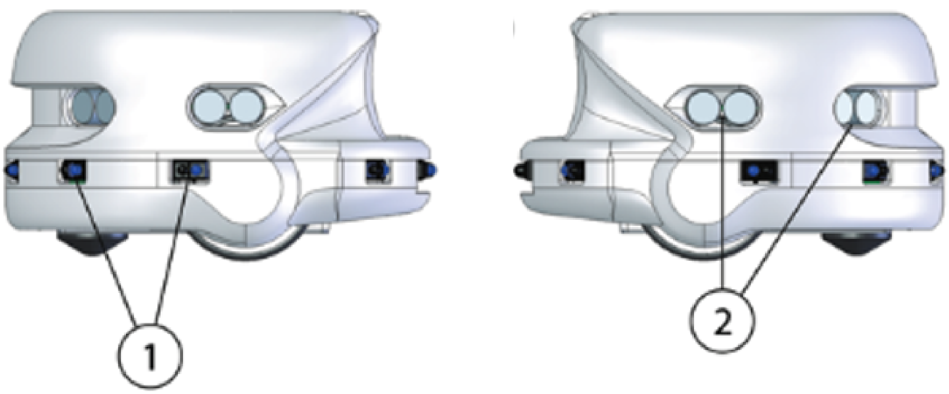


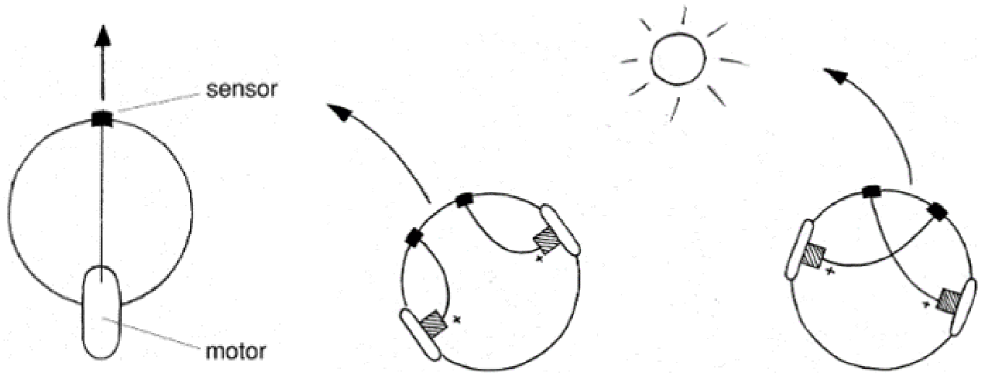
Figure 4.2: Position of the sensors used in the robot. (1) Infrared sensors. (2) Ultrasonic sensors (Adapted from Figure 3.1 of the Khepera III user manual [Sch+08])

robot will be explained. The implementation of this method makes the robots to explore the environment without colliding, as well as avoiding obstacles and other robots.

The design of these vehicles allows the robot to respond to changes in their environment [Wer91]. The vehicle designer is free to select sensors attributes, number of neurons, for the motors speed calculation process, and environments. In order to start the explanation, we are going to examine the simplest Braitenberg vehicle of the series.

As shown in Figure 4.3a, the first Braitenberg vehicle has one sensor, for one particular quality, and one motor. The sensor and the motor are connected very simply: The more there is of the quality to which the sensor is tuned, the faster the motor goes. If this quality is temperature, it will move fast in hot regions and slow down in cold regions. An observer might get the impression that such a vehicle likes cold and tries to avoid heat. The precise nature of this quality does not matter; it can be concentration of chemicals, temperature, light, noise level, or any other of a number of qualities. The vehicle always moves in the direction in which it happens to be pointing [Pfe+01].

The second vehicle 2 is very similar to vehicle 1, except that it has two sensors, one on each side, and two motors, right and left (Figure 4.3b). There are two possibilities for connecting the sensors to the motors (Figure 4.3b). The left case shows, the sensors are connected in the same line of the motors, the obtained effect of this connection is



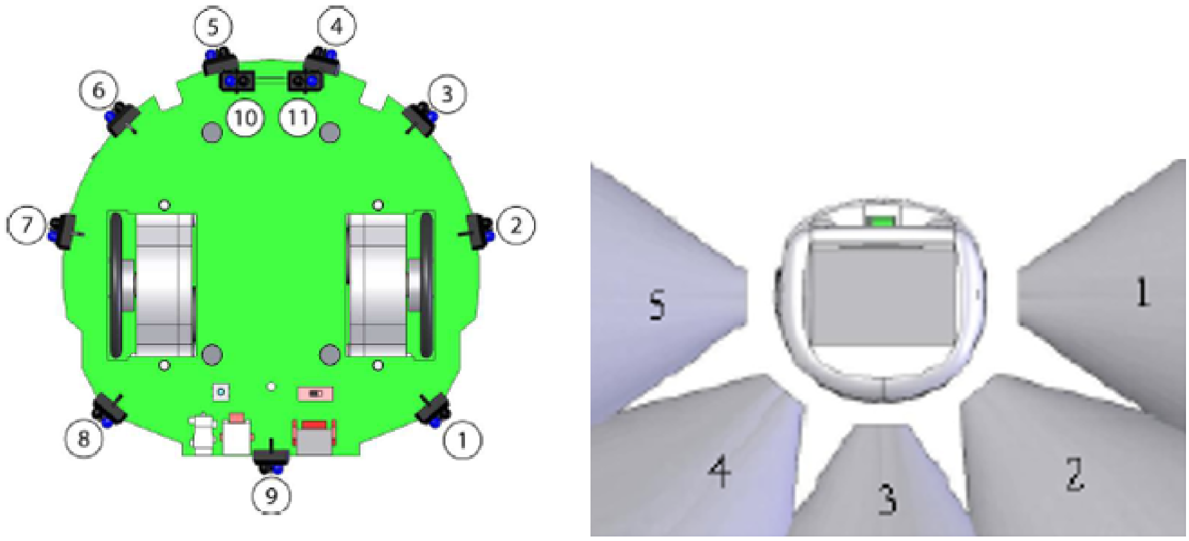
(a) Braitenberg vehicle conformed by one sensor and one motor.

(b) Braitenberg vehicle conformed by two sensors and two motors.

Figure 4.3: Example of Braitenberg vehicles conformed by the same amount of sensors as motors. (a) A sensor controls the speed of the motor. Motion is always forward, in the direction of the arrow, except in the presence of perturbations, like friction. Figure taken from [Pfe+01], Chapter 6, page 183, Figure 6.1. (b) Vehicles in the vicinity of a light source. The left robot is programmed to orient itself away from the light source. The right robot is possess the same program as the previous one, except that the cables that control each motor are interchanged. This provokes the vehicle to orient itself towards the light source. Figure taken from [Pfe+01], Chapter 6, page 184, Figure 6.3.

the robot moving away from the source of light. This behavior is produced because the right sensor of vehicle is closer to the light source than the left, it gets more stimulation and thus the right motor turns faster than the left. However, when the cables are switch as in the right case, the robot is attracted to the source. As outside observers, we might characterize the vehicles as follows: Vehicle (Figure 4.3b, left) is a coward, whereas vehicle (Figure 4.3b, Right) is aggressive.

The "brains" of these vehicles are very simple. They consist merely of two neurons connecting the sensors to the motors. Note, however, that seemingly complex interac-



(a) (b)

Figure 4.4: (a) Bottom view of the Khepera III showing the positions of the IR sensors. Figure taken and Modified from [Awa+18], page 11, Figure 1. (b) US sensors position in the Khepera III. Figure taken from [Luv+18], section 3.2 page 12, Figure 5.

tions among these vehicles can emerge. In order to see such interactions we offer the resulting videos of the research [VideosResult]

4.2.1 Khepera III using a Braitenberg Inspired Model for Obstacle Avoidance

The Khepera III is a small, circular mobile robot running on two wheels. The diameter is about 130 mm, the height about 70 mm [Sch+08].

In its basic configuration, the Khepera III is equipped with two motors with associated controllers, a ring of 9 infrared (IR) sensors attached to the bottom layer of the robot's internal structure (see Figure 4.4a), another ring of ultrasonic (US) sensors attached to the second layer (see Figure 4.4b).

As explained in the last section, the Braitenberg vehicles programmatic model can be represented as two neurons connected to the motors. This model resembles a simple

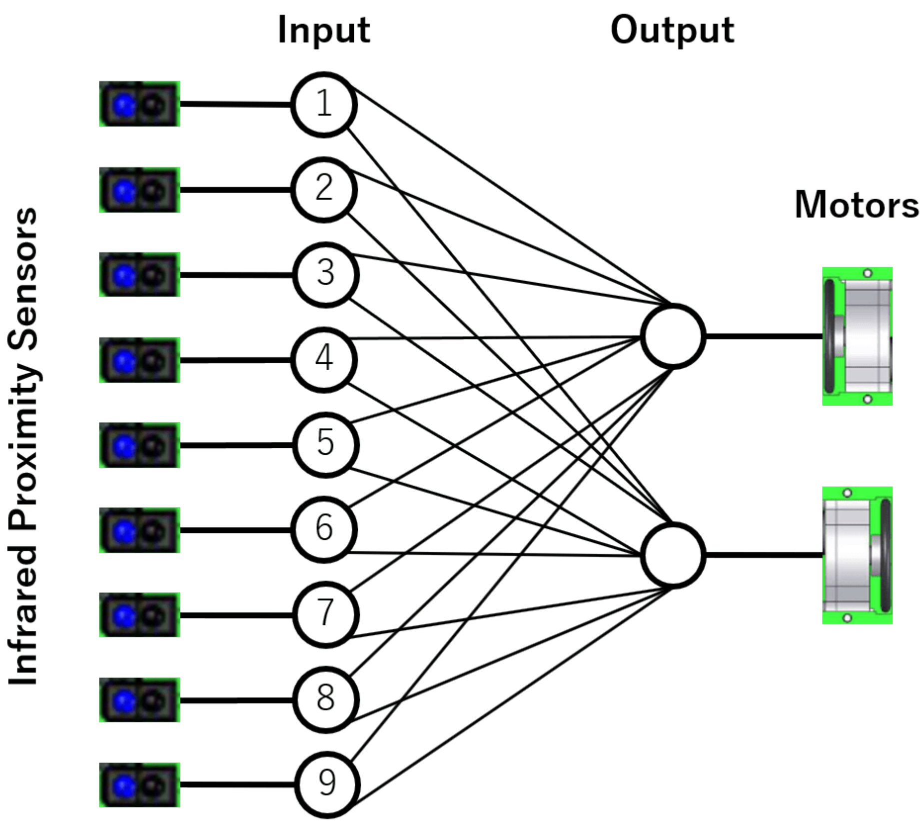


Figure 4.5: Representation of the simple WNN which models the behavior of a Braitenberg vehicle with nine sensors (input) and two motors (output). When the information of the sensors is received after a step of time pass, each value would be multiplied by the corresponding weight in order to inhibit or excite the respective motor. The weights are determine to change the behavior of the mobile robot into a coward (orienting itself away from the source, the distance from an obstacle) or aggressive (orienting itself towards the source).

weighted Neural Network (WNN) [Isl+17]. In Figure 4.5, the connection between the inputs, represented by the 9 infrared sensors, and the outputs (the two motors) are graphically presented.

The result of the output neurons is the speed of the left and right motor of the Khepera III in *rads*/*sec*. Each of the sensors' readings is multiplied by the value of its

	Motor 1	Motor 2
IR1	-20000	40000
IR2	-30000	50000
IR3	-70000	60000
IR4	60000	-70000
IR5	50000	-30000
IR6	40000	-20000
IR7	-5000	-5000
IR8	-10000	-10000
IR9	-5000	-5000

Table 4.1: Neural Network weights used for the Braitenberg model 4.5 in the Khepera III using VREP. The positive values are those that excite the corresponding motor. On the other hand, the negative values are those that inhibit the associated motor. All these factors are summed to obtain the motor speeds to move the mobile robot.

neuron connection weight. This values are aim to produce a mobile robot characterized as a coward. As a result, if the robot is close to an obstacle or another robot, this will tend to move to the opposite direction. The result of this process is the so called obstacle avoidance using the Braitenberg vehicle model. The weights of the WNN using in the Khepera III simulation environment are shown in table 4.1.

4.3 OBP Implementation using VREP's Khepera III Sample

In order to perform the first test of the OBP algorithm we used the following sensors:

- 9 infrared distance sensors
- 1 camera with 3 simple color filters

As long as the environment was open and the illumination for the camera was low, the camera filtered poorly the colors and the surrounding of the environment. Therefore, we installed a spot-light in the center of the simulation environment.

4.4 Khepera III Implementing The Guide-Based OBP Method

For the following experiments, we had to make minimum extension in the usage of the robot sensors given that our proposed method involve a new type of agent called guide robot. This one can be any robot withing the swarm, but its color might change depending on which task it decides to take. To see the colors the robot can change, refer to Figure 4.6

- 9 infrared distance sensors
- 1 ultrasonic sensor (us3, the middle sensor out of five)
- 1 camera with 4 simple color filters

The robots can recognize four colors in order to differentiate the targets and other members of the team. The following colors are described by how the robots recognize them. The first two (red and blue) are intrinsically associated with the Occlusion-Based Pushing method (refer to section 2.2). The other two colors (green and orange) were added in order to achieve the realization of the proposed Guide Mode to work along with the traditional OBP:

- Red: Home detection.
- Blue: Object detection.
- Green: Normal robot detection.
- Orange: Guide robot detection.

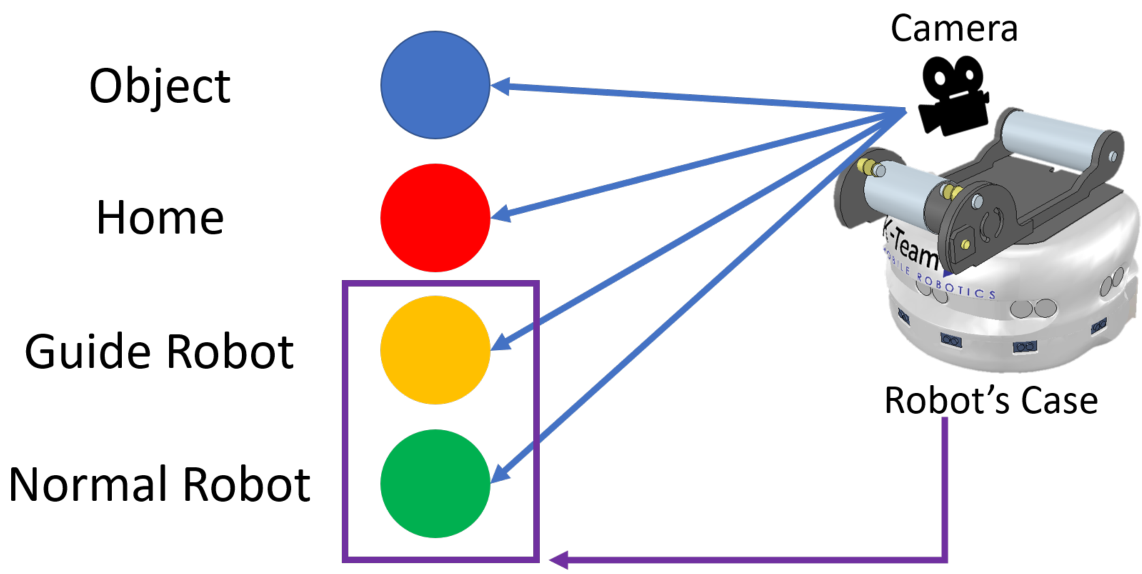


Figure 4.6: Color detection by simple camera filters. The camera is placed on top of the robot and four color filters are implemented. Each filter is in charge of taking the color feature out of the raw image data. Finally the filtered data are transferred to the robot in order to take decisions regarding the conditions stated by the proposed algorithm.

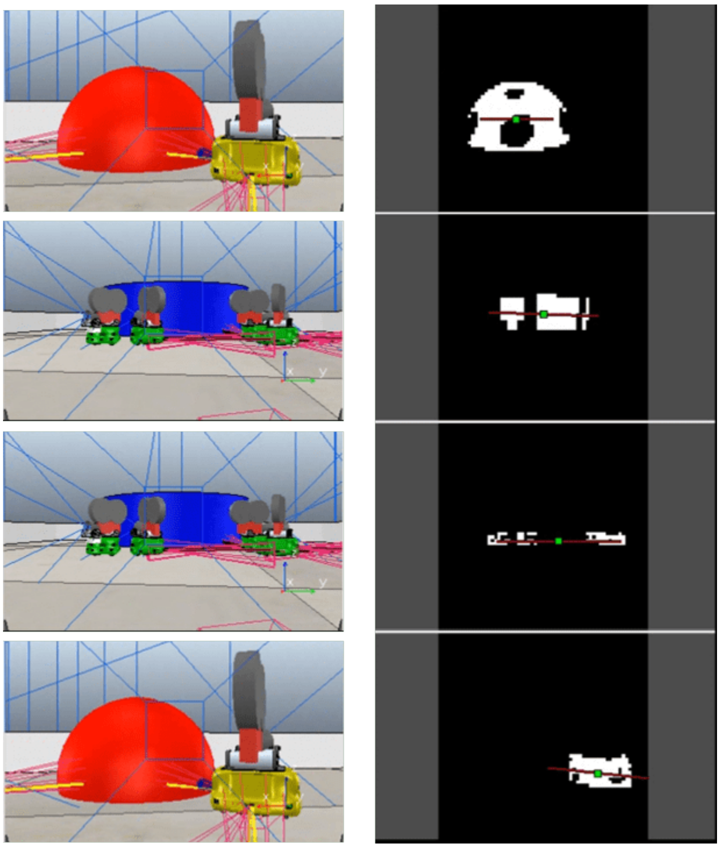


Figure 4.7: Color detection by the camera on top of a robot. Left side, raw camera information received by the robot. Right side, colors filtered by the program taken from the raw data of the left side. After the data are filtered, it is also given a feature to measure the position of the center of mass of the color perceived. This feature can be seen as a green dot centered in a red line. Note: on the right side, the colors are viewed in the following order from top to bottom: red, blue, green and orange.

Chapter 5

Implementation of a Self-Guided Swarm of Robot for Cooperative Transportation

To evaluate the effectiveness of our approach, we built a novel dataset containing multiple pairs of first-person videos and points-of-gaze data. To the best of our knowledge, this dataset is the first to use multiple points-of-gaze sources in first-person vision tasks. The experiments demonstrate that our approach can outperform several state-of-the-art commonality clustering methods on the task of discovering objects of shared attention in various interaction scenes.

5.1 Experimental Set-Up

To assess the guide mode algorithm to work aside with the OBP in a 2-D environment, we implement a decentralized swarm system. This means, each robot is programmed with the same algorithm developed in Section 4. No human manipulation is necessary in the accomplishment of the tasks presented.

The environment design is based on the work of Chen et al. [Che+15]. Nevertheless, the distribution of the elements in the environment change, including the number of

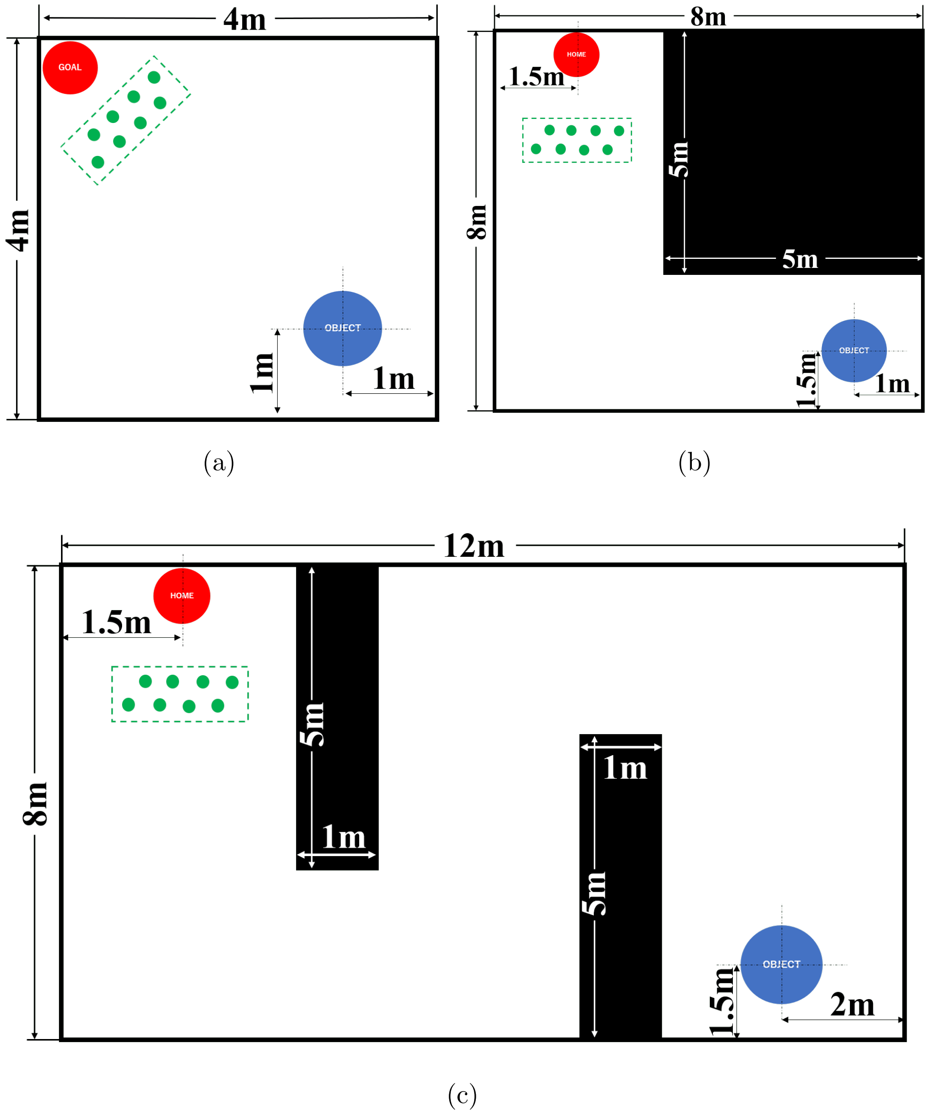


Figure 5.1: Environments for the Experiments. (a) Environment 1: OPB experiment having as Goal the red circle and the object to transport the Blue circle. (b) Environment 2: First experiment of the Guide Mode having as Home the red circle. (c) Environment 3: Second Experiment of the Guide Mode.

robots. In Figure 6, we can observe the different environments used for these experiments.

For the first experiment (Figure 5.1a), we have an environment of 16 m^2 of working area with the object placed in the bottom-right corner of the area. The goal is located in the top-left corner of the area in order to perform the OBP method.

In Figure 5.1b, we enlarge the environment to 64 m^2 forming an obstacle of 25 m^2 resulting in a total work-area of 39 m^2 with walls of 0.2 m high. Here, we tested the robots' decision making, when the OBP method cannot be performed on the first place.

The last environment (Figure 5.1c), possess an external area of 96 m^2 and a total working area of 86 m^2 with walls of 0.6 m high that occupy 10 m^2 of the total area, serving as obstacles for the task. Here, the Guide Mode algorithm is tested in a larger area, in order to observe how the robots, produce a path between the object and home.

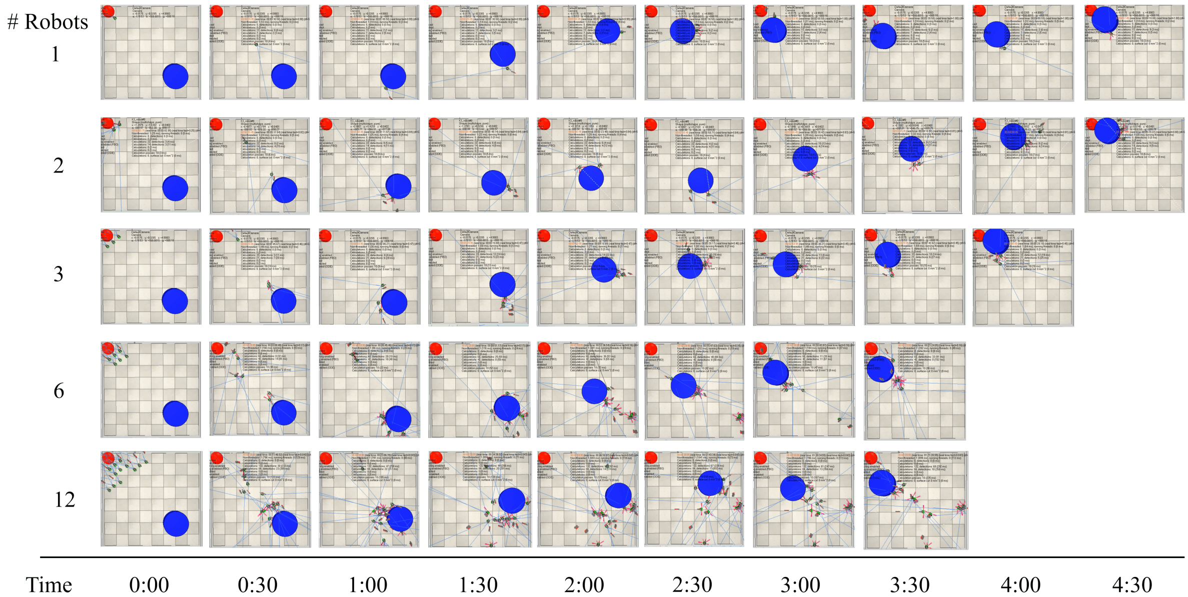


Figure 5.2: Experiment 1: OBP algorithm implemented in the group of robots. The graph is divided by 5 rows, each representing the number of robots used in each group. The evolution of the object moving though the environment is shown using a 30 seconds time-lap. The snapshots finishes in the time-lap when the object touches the goal target.

5.2 Experiments

5.2.1 OBP Method Implementation

The OBP method was implemented based of the diagram presented in Figure 2.3. This means that, it is our own application of OBP in the Khepera robots [Cor] for V-REP simulator [Rob].

In Figure 5.1a, it is shown the environment used for this experiment. We formed five groups of robots: 1, 2, 3, 6 and 12 robots respectively. For each group, we performed five simulations to get the group's best times after completing the task for transporting the object to the goal. The object has a mass of 1 Kg for all the trials in Experiment 1.

In Figure 5.2, we show the snapshot of the Experiment 1. The time format is minutes: seconds with an interval of 30 second between each snapshot. Each horizontal snapshot corresponds to a different group of robots performing the OBP method. The number of robots is shown in the left side of the figure.

In each group, five out of five trials were completed, but some simulations took more time to accomplish than others. In Figure 5.11, we show the best times of each group. The best performance among the groups of robots is the one with six units. This is due to the number of robots that are constantly working in order to push the object to the destination. In the case of the group of twelve robots, they tended to interrupt each other because it is not implemented a resting algorithm. This would allow the robots to determine when they don't have space enough to collaborate in the pushing process and then explore the environment until another time when they can be necessary. For a matter of time limitation, this resting algorithm will be a future work to implement.

We can observe that; even though, the environment is larger than the one used on Chen's experiments [Che+15] and the mass (less than 0.5 Kg) of the object is bigger, all the groups of robots finish the task in less than 5 minutes. These differences in capabilities is due to the robots used. Chen [Che+15] use the E-Puck [EPF] mini-robot and we use the Khepera III [Cor]. Our robot is approximately 65% more powerful regarding the physical characteristics compared to the e-puck. Due to this, we increased the mass of the object to transport and increase in 20% the working area.

These changes on the environment and the object mass, do not change the effectiveness of the OBP in order to push the object towards the goal. Moreover, the success on the implementation of the OBP in V-REP gives us the opportunity to make the experiments for the guiding process in larger and complicated environments.

5.2.2 Adding Self-Guided OBP in a Unicursal Environment

In a larger and complicated environment, the OBP method would be stuck at the "Search Object" (refer to Figure 2.3). This is due to the impossibility of perceiving the object in an environment that is not open. For instance (Figure 5.1b). The environment we design for our first experiment testing the Guide Mode, has 39 m^2 of total working area and 25 m^2 that block the direct view of the object from the robots' point of view. Consequently, the robots will not be able to locate and approach to the object inside of the environment. The object has a mass of 2 Kg for all the trials in Experiments 2 and 3.

In order to overcome the case mentioned above, we designed a first state called "Check Home & Object" as Figure 3.2 show on the top of the state diagram. Therefore, if the robots will detect the object and home in this initial state, they would perform the traditional OBP method (bottom left state of the state diagram in Figure 3.2). Otherwise, if the robots just detect in the initial position the location of Home, the robots are meant to perform the Guide Mode algorithm explained in Figure 3.3. The algorithm starts with the explore for object state until the object is found. Consequently, the robot should check if there is any guide or Home close to its position. In an affirmative case, the agent will perform the traditional OBP method having as sub-goal the guide or Home detected. On the other hand, this robot will turn back to look for the most suitable position for place itself in a certain distant from a previous guide or Home.

We realized five simulations using the environment described above and selected the three best results, in order to show the effectiveness and reliability of the proposed self-Guided OBP method. The first experiment presented was performed with six robot.

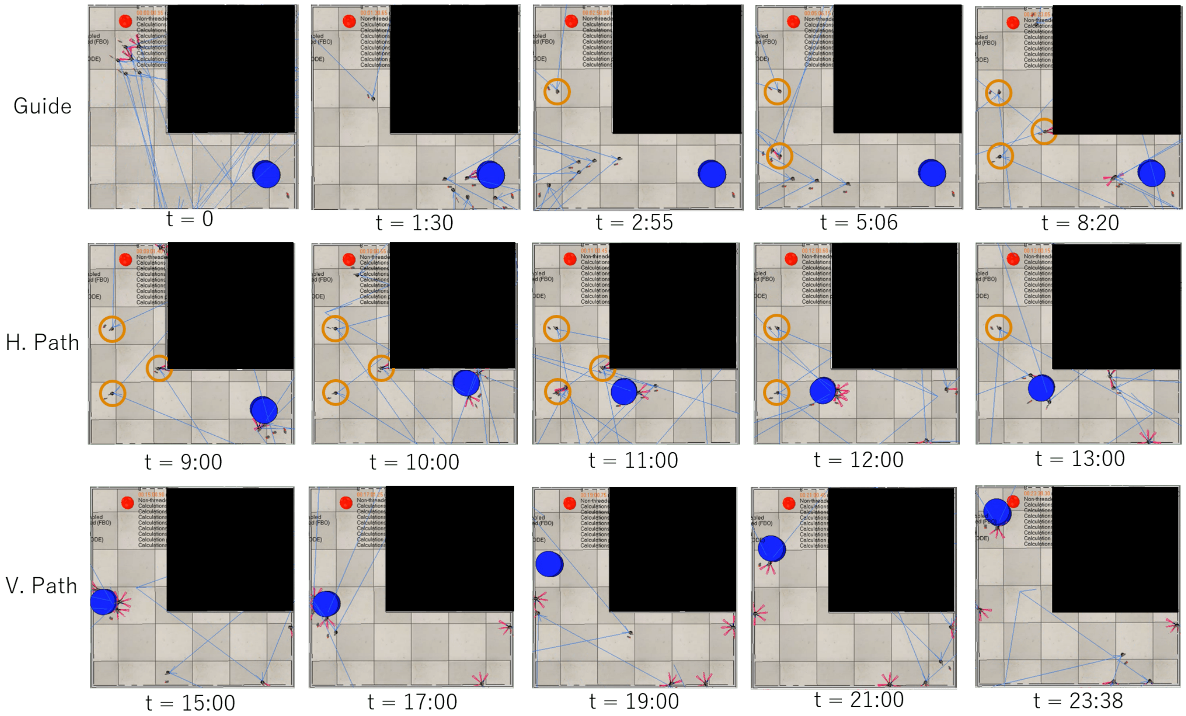


Figure 5.3: Experiment 2.a: Six robots programmed with the self-guide algorithm (Guide Mode) and OBP method. The orange circles represent the location of the guide robots during the simulation.

From the Experiment 1, (section 5.2.1) we determined that the optimal amount of robot, able to perform the task of retrieving an object faster in an open environment, were six units. The motivation of using the same number of robots in this new environment is to determine, if the same amount of robots performing only OBP can perform the task of guiding themselves to retrieve the object back to Home.

Experiment 2.a: Six robots applying the proposed method.

The snapshot shown in Figure 5.3 displays three rows representing the important parts of the object retrieval process. Below each snapshot, the time the scene was taken in minutes : seconds. The first row represents the process and the time needed to complete the Guide Mode after the robots realized, the object wasn't visible from their initial position. The second, shows the robots pushing the blue object through the horizontal

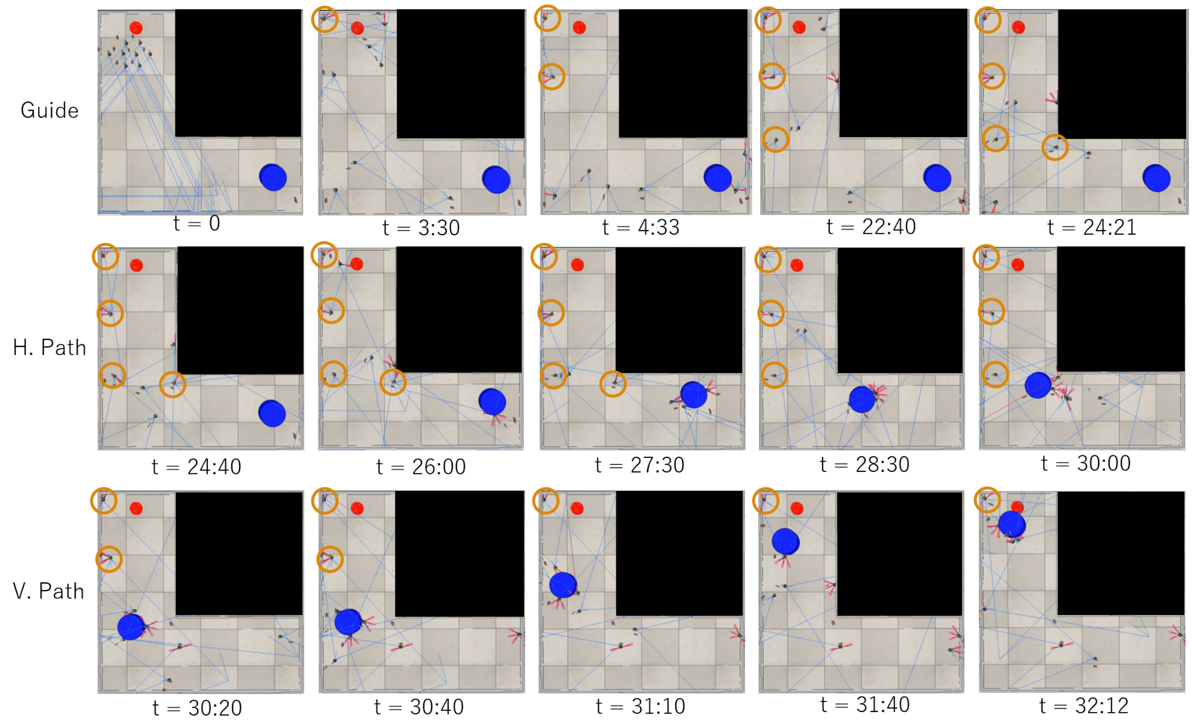


Figure 5.4: Experiment 2.b: Eight robots application with the Guide Mode for OBP. Here we can notice that the robots (represented by orange circles) form the shape of the free space of the environment.

path of the environment. The last row displays the robots pushing the object through the vertical section of the environment until they finally reach Home.

The simulation finished in 23 minutes, 38 seconds (exactly the time when the blue object touch the red target, Home). The self-guiding process took 8 minutes, 20 seconds to conclude and let the pushing process to start. The pushing process begins right after the robots, proximate the object, realize were to go by perceiving the presence of the last guide.

Experiment 2.b: Eight robots applying the proposed method.

The next simulation (Experiment 2.b), is done in the same environment. However, we increased the group of robots up to 8 in order to test how scalable is the swarm with the proposed Self-Guided OBP method. Figure 5.4 shows the snapshot of this experiment.

The simulation last for 32 minutes, 12 seconds, which was not the expected results after increasing the number of robots. The guide process explains why the simulation lasted longer. In the first row (Figure 5.4) between the 3rd and the 4th, we can see that there is a difference on time of approximately 18 minutes, 10 seconds. This differences is due to the existence of other robots in between the second guide, placed at t = 4:33, and the robots able to be guides, making them turn to look for a better position without other agents in the middle of the line of sight of the previous guide. However, the time used by the rest of the robots to push the object (approximately 7 minutes, 32 seconds) was shorter than in Experiment 2.b (14 minutes, 38 seconds).

Due to the explained above, we run 3 more simulations in order to determined the best times obtained with 8 robots on this “L” shaped environment. Although, the situation regarding other robots running into the line of sight of the possible guides continued. Nonetheless, the simulations were getting shorter in average.

Experiment 2.c: Best time of eight robots applying the self-Guided OBP method.

The last simulation shown in this section (Experiment 2.c) is the result with the faster performance taking into account the time necessary to complete the self-guiding and the pushing process.

In Figure 5.5, we show the snapshot of the best result out of five simulations (counting the Experiments 2.a and 2.b). The best result is determined by the shortest time simulated to accomplish the task of retrieving the object back to Home. The whole experiment last 19:20 (min:sec) divided in the guiding process and the pushing process. It is important to remember that, the robots would not start the pushing process, until they know where to push.

We can observe that, the time of the last guide is place (at the end of the guiding process) is 8:25 (min:sec), leaving the rest of the task time to the pusher robots. When the object is close to one of the guides (refer to Figure 9 in time 8:25 (min:sec)), this one becomes a pusher that helps the rest in the process of retrieving the object.

According with these results, no robot has an specific work to do in the process

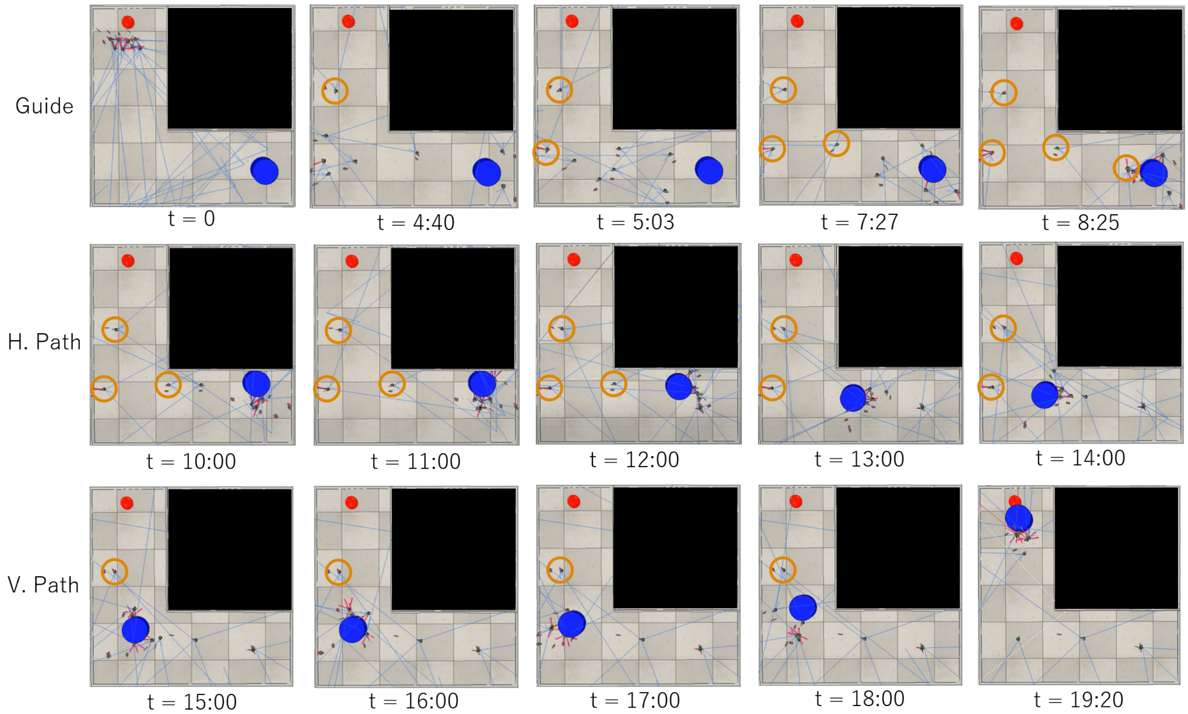


Figure 5.5: Experiment 2.c: Eight robots simulation with the best time obtained, to complete the object retrieval using self-Guided OBP method. The guide robots also formed the shape of the free space of the environment (See the orange circles). This is the fastest simulation out of three performed with eight robots.

of retrieval. Therefore, regarding the situation they find themselves, the robot decides whether to be an explorer, guide or a pusher. Consequently, this behavior shows that the proposed self-Guided OBP method is autonomous and robust. Finally, we tested our method in a larger and complex environment to confirm the reliability and the time needed to finish the task.

5.2.3 Self-Guided Pushing in a Unicursal Complex Environment

The last experiment with successful results presented in this research is performed in the environment shown in Figure 5.1c. It has a working area of 86 m^2 where the robots explore looking for the best spot to guide the team to the final destination (Home) with the object. The amount of robots used is 16 unit, this number have been estimated accordingly with the results of the Experiment 1 (section 5.2.1) that give us and idea of the quantity of robots to use per square meter 5.3.

In Figure 5.6, we can see the snapshot of the complete simulation. This is 1 out of 3 simulations done, having this one as the best trial taken. It has a duration of 2:40:00 (h:min:sec), where it took for the guiding process 1:15:22, leaving the rest of the time to the pushing process.

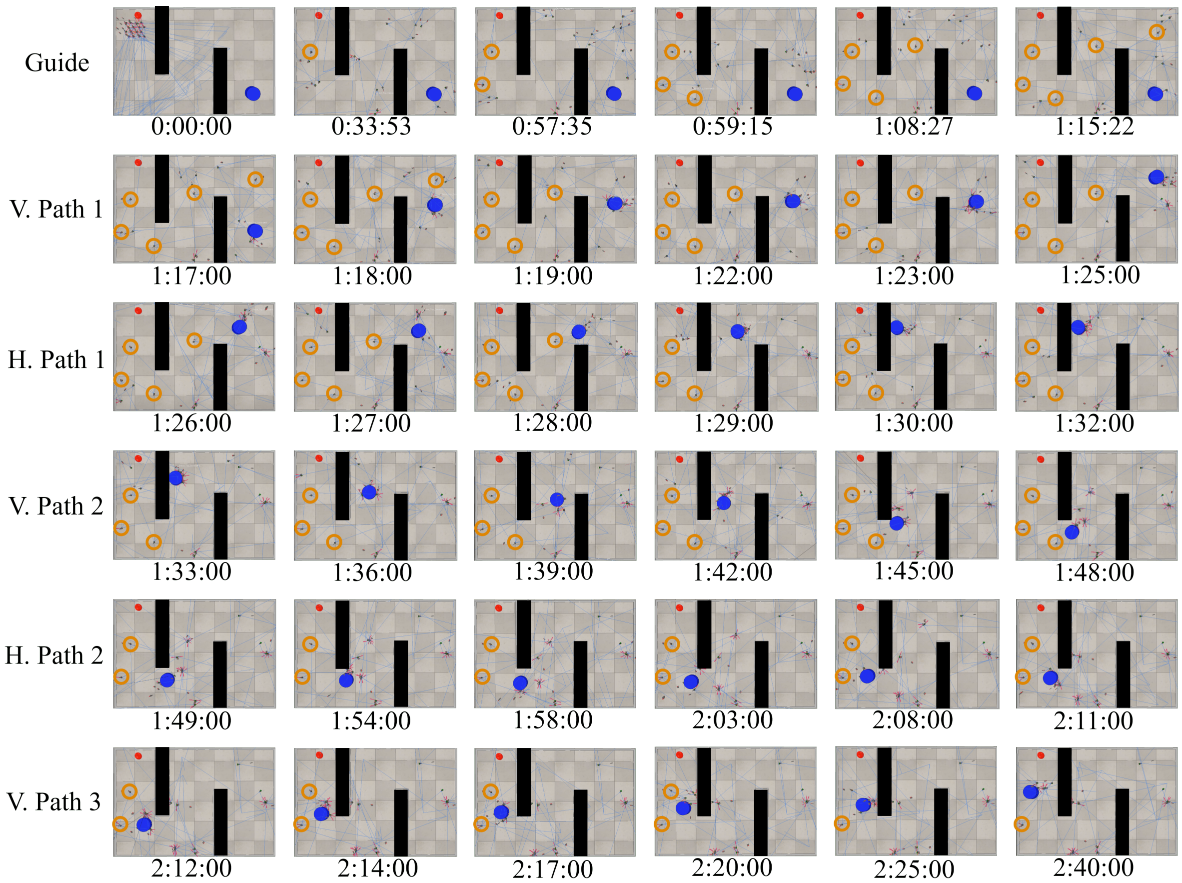


Figure 5.6: Experiment 3: self-Guided OBP proposed method applied in a large environment. The environment is unicursal, in other words, the path of the environment is only one and can be traced with an interrupted line. The guides were placed (see orange circles) by themselves optimizing the amount of robots to do such task.

The snapshots are divided by six rows, representing the important parts of the object retrieval process. Below each snapshot, the time the scene was taken in minutes : seconds. The first, row represents the process and the time needed to complete the Guide Mode after the robots realized, the object wasn't visible from their initial position. The rest of the five rows are sections of the environment divided as follows: three vertical paths ("V. Path") and two horizontal paths ("H. Path"). Here the guide positions (marked with an orange circle) were successfully identified by the pusher robots.

These experiments were the longest ones regarding the simulation time and the rendering time by the simulator. Due to the amount of robots, the simulator V-REP took three and a half days to produce three hours of simulation on real time. Working with simulation with complete physical models is computationally expensive, and more when the number of agents increase. The renderings and the calculations of the movements of each robot are the factors that multiply the simulation time. Therefore, the application of our method with real mobile robots would accelerate the experimentation process, hence the results would be more in number and in reliability.

Moreover, the simulations in Experiment 3 lasted longer, due to the fact that the new guides needed a clear view to the previously placed guide or Home. Which means, in the line of sight of the potential guide, no robot (green colored robot) should exist in order to make a suitable placement. The large amount of robots on simulation did this task very difficult for the first two guides (Figure 5.6), due to the accumulation of robots in the Home zone within the exploration time. Nonetheless, in the end, the task was successfully accomplished after several iterations of the robots exploring the environment.

These results demonstrate that, the swarm of robots with our method accomplish the task of retrieving an object in a complex environment autonomously and without a leader. Therefore, we can summarize that our algorithm works for areas with obstacles or walls that define an unique path to take. However, it is also interesting for us, to test our algorithm in other types of environments designed to have different tracks which might not lead directly to Home 5.2.4.

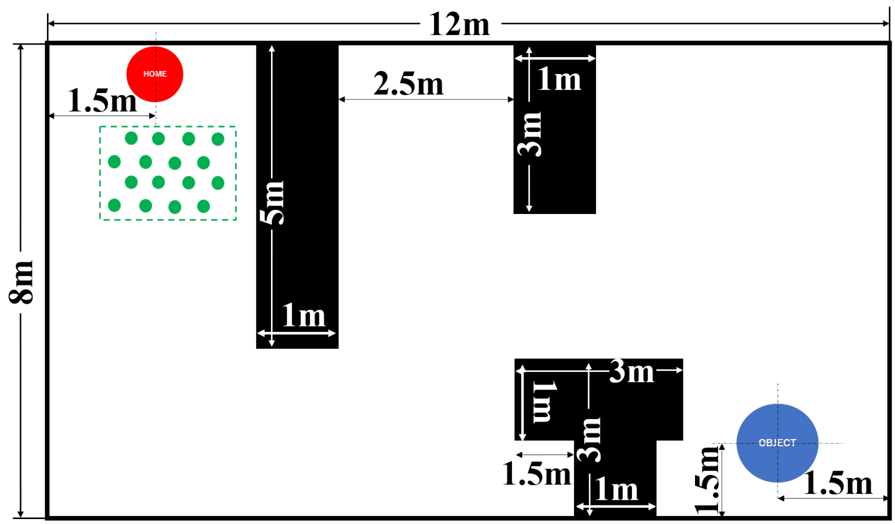


Figure 5.7: Experiment 4: Environment design to test the proposed method in an environment with a branched path. These type of environments are called maze-type, they are defined as, areas composed by obstacles that do not form single paths, but have multiple routes from the starting point that might lead or not to the target (find the object and retrieve it).

5.2.4 Self-Guided OBP applied in a Maze-type Environment

Knowing that the proposed self-Guided OBP successfully performs the object retrieval in single track (unicursal) environments, it is natural to conclude that the interior of collapsed buildings or hazardous areas are not simple places to explore. Clearly, such areas are intricate having routes in many directions that would not ensure an easy return for robots exploring these surroundings. These environments are called maze-type areas.

Therefore, the Figure 5.7 was designed to test the proposed method in a maze-type environment to determine how the guide robots manage to trace the optimal path back to Home (represented by the red spot on the left of the Figure 5.7).

The first simulation (Experiment 4.a) ran within this environment lasted for three hours, being equivalent to three day and a half of rendering and calculation time from VREP simulator. In Figure 5.8, we can observe a short snapshot which represent the first 35 minutes of the simulation process. It is shown a small portion of the simulation, due

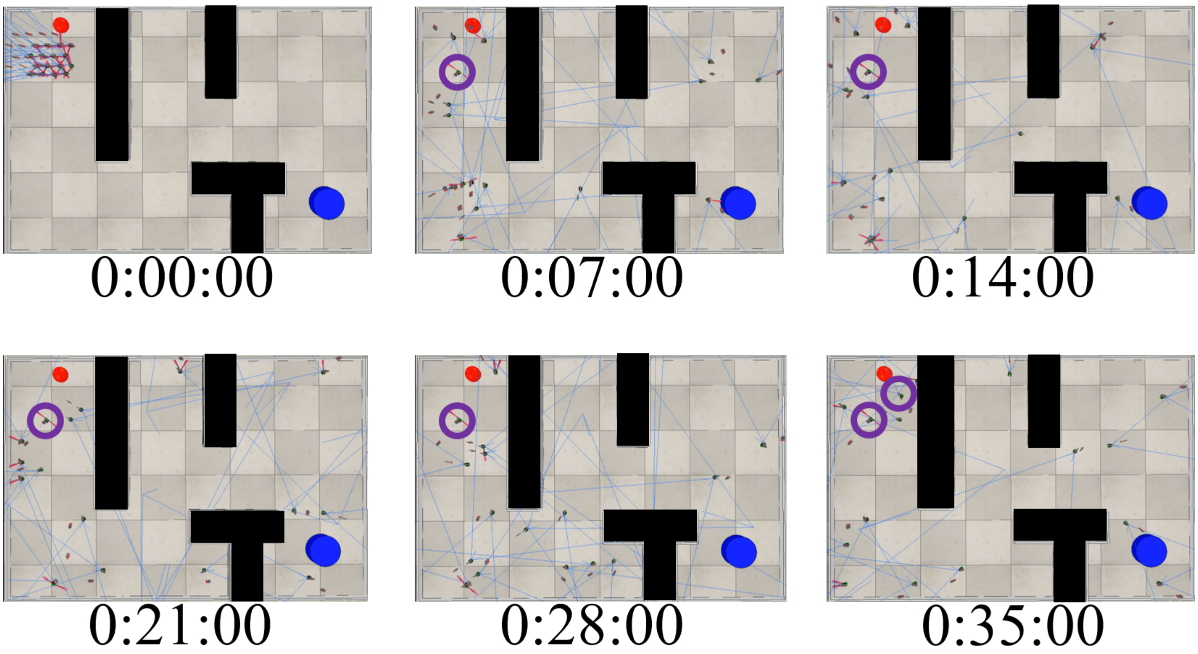


Figure 5.8: Experiment 4.a: Maze-type environment for self-Guided OBP method results. The purple circles represent the bugged robots that where flipped by other agents while exploring the environment. The fact of having flipped robots is not a problem because agents with bugs can stop their task and it can be replaced by others. The problem lies in the proximity these robots have to the home or a guide position (refer to Experiment 2.b and Figure 3.3)

to not obtaining any result from the self-Guiding process and consequently, no results from the pushing method as well.

Here we can observe that after 7 minutes of simulations passed, a robot got flipped by other one living it unable to pursue any movement. The circumstance of having a flipped robot is not important in a decentralized system of multi-robots, because other individuals would be able to replace the bugged robot. The vulnerability of this situation is that, this bugged robot is in front of the Home. Therefore, if we refer to Figure 3.3 on the right side, we would find an important condition for the self-Guided OBP to work. It says that, if the potential guide robot finds the localization of Home or another guide but at the same time, there is another robot in front of its line of sight, the agent should abandoned its aim of being a guide. The explanation behind this behaviour is due to the assumption that, the robot in front of the potential guide might have the same aim. Consequently, it can avoid the accumulation of multiple guides in the same spot. However, we can see that it can be a strong limitation for the proposed method, if it is not applied the resting algorithm explained in section 5.2.1.

Fortunately, the second simulation (Experiment 4.b) produced a better behaviour than the expected initially. However, the recording of this second simulation was corrupted by the simulator environment. Therefore, it cannot be seen the complete development of the four days worth rendered simulation. As a notice, the recording of this or the previous recorded simulations were not cut. The only edition made was the fast-forwarding in order to fit the length of the DEMO in the presented time [Lan].

In Figure 5.9, we can observer in the first row, "Guide", the process taken by the robots in order to place themselves as guides. This "first" guide process finishes within 01 hours, 50 minutes and 11 seconds. However, this snapshot's time is result of the corrupted recording obtained from the VREP simulator. In order words, the process previous to the result of the snapshot in the time 1:50:11 could not be recorded due to a problem with the simulator. Similarly, the next and final snapshot, of the guiding process (1:50:44), it is derived from the same problem. Hence, in order to explain with more details the process that took place in this simulation, a graph will be presented with the estimation of the position of the guides to produce the observed movement of the object (Figure 5.10).

In Figure 5.10, it is represented the estimation of the placement of the guide robots during the times the simulator did not recorded continuously. This estimation is possible due to the position of the object in Figure 5.10 time 1:50:11. The object was stuck in the corner where possibly the pusher robots perceived the last guide of the first path formed. However, after the last guide started to help to the rest of the pusher team, they could not be able to perceive the next guide due to its long distance from them. This situation made the agents to push the object until the following wall in front of

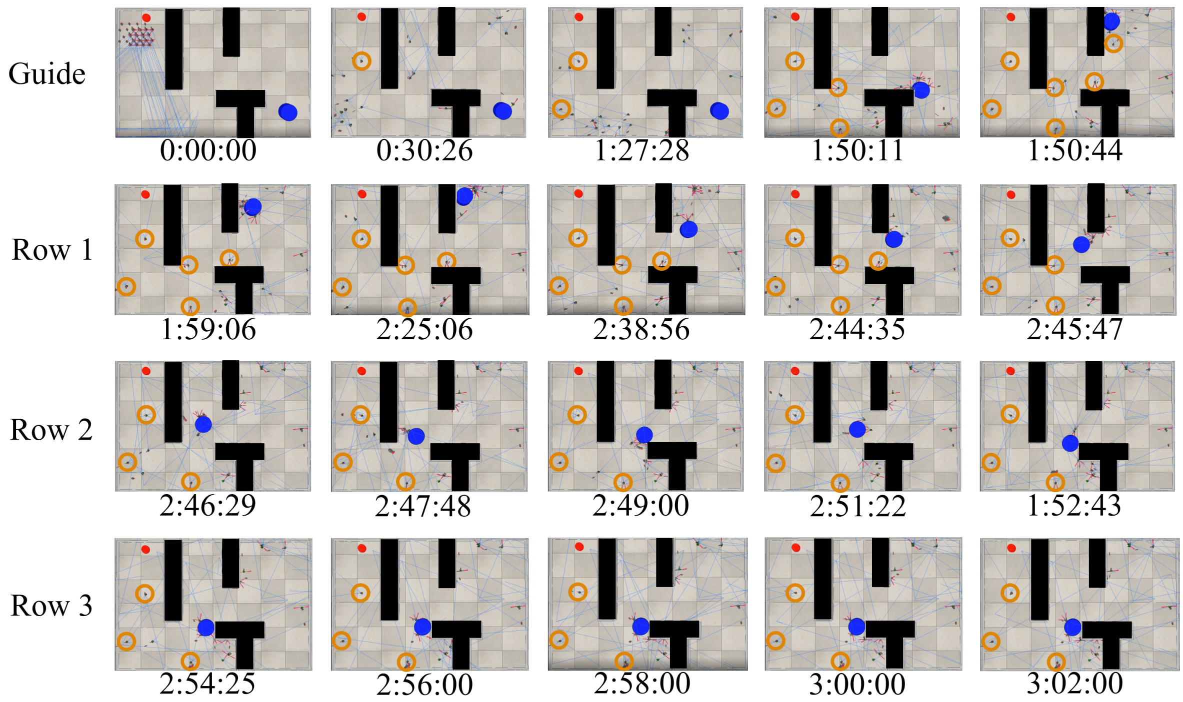
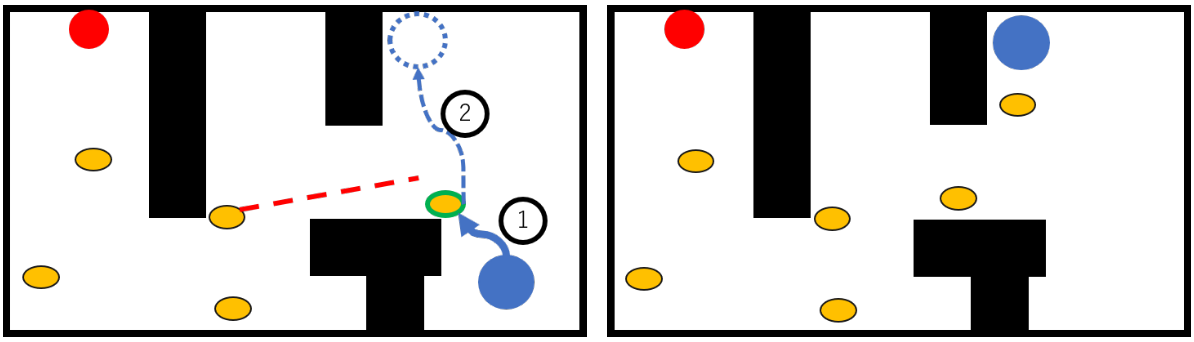


Figure 5.9: Experiment 4.b: Maze-type environment for self-Guided OBP method results. The first row of the figure, "Guide", shows the times when the robots took place as guides. We should remind that the recording of the simulation got corrupted. Therefore, parts of the guiding process, as well as, the pushing process were not save in the recording file. Nonetheless, In the rows 1 and 2 can be observed the pushing process from the top of the environment. Row 3 shows the exact time when the agents stopped their movement, due to the long simulated time. Certain incremental values, such as the encoder of the motors, overflowed and stopped the movement of the last robots pursuing their tasks.

them.

The following snapshot in Figure 5.9 (1:50:44), shows the reconfiguration of the self-guiding robots. The new organization is obtained after an undetermined period of time. The robots were repeatedly going to the object and the next guide in order to establish a new location for themselves with respect to the new position of the object that did



(a) Previous path estimated. (b) New path self-reorganized.

Figure 5.10: Path reorganization to overcome a group mistake. (a) Shows the estimated organization in order to find the object in the corner of the orange-green guide (circle one). The red dashed line, indicates that no agent could detect the next guide due to its far distance to the rest of the pushing team. Circle two shows the path the object took in order to be found as it is visible in Figure 5.9, last snapshot of the guide mode (1:50:44). (b) Represents the reorganization the robots made, in order to place themselves as guides regarding the new position of the object.

not follow the previous configuration shown in Figure 5.10a.

This behavior demonstrate how far the proposed algorithm can go, if a long period of time is provided to the robots to work. Moreover, the self-Guided OBP method have shown that without any modification to its base algorithm, it is possible to produce a swarm of mobile robots that can correct their own path if any mistake is made by the group. This behavior is observable in nature, having as an example an ant colony searching for food.

5.3 Results

In Experiment 1 (section 5.2.1) we have determine the optimal amount of robots to work in an open environment with the OBP method, that is approximately 0.375 bot/m^2. By calculating the amount of robot for each larger environment, we realized that it would be computationally expensive to simulate such large groups of robots on Experiments 2 and 3 (section 5.2.2 and section 5.2.3 respectively). Therefore, the calculated number of robots by this factor will be divided by 2, which results in more than the minimum amount of robots necessary to accomplish the tasks (new area-robots factor 0.188 bot/m^2).

As it is visible in the previous section the Guide Mode was successfully done but depending on the environment, the time use in order to accomplish the task would be longer or shorter. In the case of the Experiment 2.c, the robots explored the environment and quickly settled the necessary amount of guides to produce a path from the object to arrive at the destination (Home). The amount of guides needed were four that placed themselves in the configuration displayed in Figure 5.12c due to the favorable situation around them. This favorable situation includes: no presence of other robots in the line-sight to Home and having enough space to maneuver.

The latter, involved the double amount of robots, which happened to be partially concentrated around home, not permitting to the robot that decided to be guide to observe properly the color of Home or other guide, introducing undesirable noise to the camera. Therefore, the chances of the robots that decided to be guides of having an opportunity to have a glance of Home or another guide without noise was small, thus the guide process took approximately 4.5 times longer than the expected, taking into account the average time of completion of Experiment 2.

5.3.1 Summary of the Experiments

The best time among all the group of robots tested applying OBP in an open environment is 3 minutes 24 seconds. The optimal group of robots able to achieve this time is conformed by six units (see Figure 5.11).

For Experiments 2 and 3 the results are going to be summarize in Table 5.1 and Figure 5.12

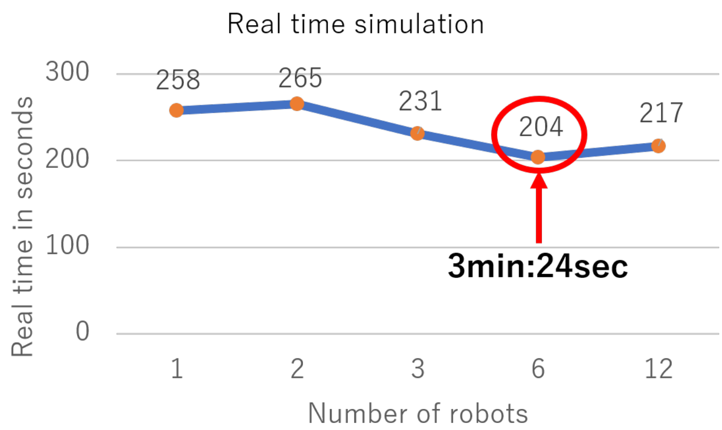
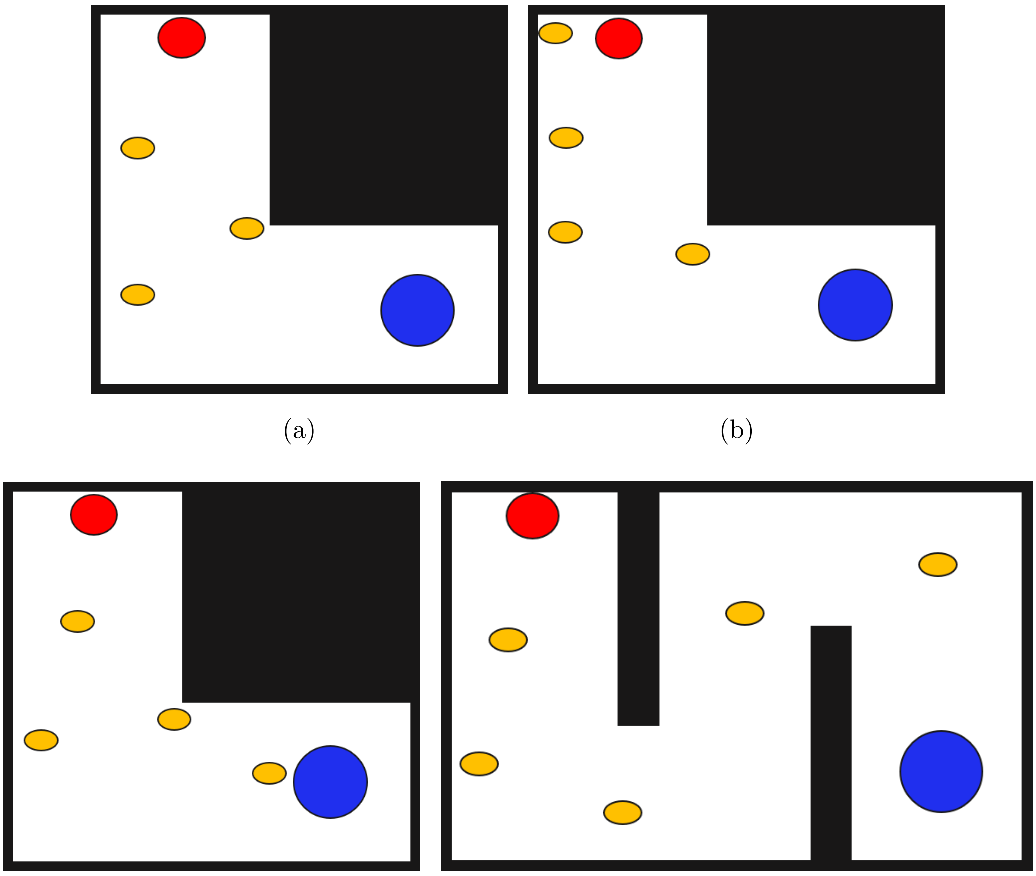


Figure 5.11: Experiment 1: Best times for each group of robots. The best time register after all the simulations was 3 minutes and 24 seconds, performed by a group of 6 robots. This shows the amount of agents necessary to successfully finish the task of retrieving an object. From these data it is possible to calculate the amount of robots necessary for the next Experiments 5 (area-robots factor 0.188 bot/m^2).

	Env. Area	# Robots	#Guide R.	Guide Mode time	Simulation Time
Exp 2.a	40 m^2	6	3	00:08:20	00:23:38
Exp 2.b	40 m^2	8	4	00:24:21	00:32:12
Exp 2.a	40 m^2	8	4	00:08:25	00:19:20
Exp 3	86 m^2	16	5	01:15:22	02:40:00

Table 5.1: Summary of the Guide Mode experiments. In this table is appreciable, the proportion of time the Guide Mode takes from the total period of the simulation. In average takes approximately 50% of the total period of the experiments.



(a) (b) (c) (d)

Figure 5.12: Summary of the Guide process experiments graphically. (a) Experiment 2.a, (b) Experiment 2.b, (c) Experiment 2.c and (d) Experiment 3. The images show the position of the robots that located themselves as guides (orange spots) in every experiment with unicursal environments.

Chapter 6

Discussion

In the field of robotics, cooperative transportation is referred as, the actions that one or several robots have to apply, in order to move an specific object from one point to another. In this thesis, apart from aiming to produce a swarm of robots that, cooperatively, transport an object from one place to another, it aims to produce a self-Guiding behavior by the robots, such as certain types of ant colonies do.

In this chapter, we would analyze the performance of the swarm of robots in each stage of the experimentation process, as well as the results obtained.

6.1 Traditional OBP vs VREP Implemented OBP

The traditional OBP method [Che+15] was implemented with real robots (e-puck [EPF]) within an open environment. It demonstrated great efficiency and accuracy in the transportation process, which lasted five minutes, no matter the object placed to be pushed. Moreover, all the robots were operative during the experiments. Therefore, the robots that were not pushing the object stayed moving randomly within the environment without colliding or interrupting the pushing process. This method we called it resting algorithm.

The implementation of OBP in VREP simulator was as well successful using different groups of robots. However, the simulator was not perfect, due to the noise integrated in the environment and misreadings in the sensors. The last, provoked the robots to present unexpected behaviours; such as, colliding with other robots or an obstacle, misreading colors in the environment and pushing the object before it was required.

Nonetheless, due to the type of robot used (Khepera III [Cor]) for this research, the simulation in an open environment were successful. Additionally, all the simulations lasted shorter than five minutes; being the shortest one 3 minutes and 24 seconds. This is owing to the physical characteristics of the Khepera III. Compared to the e-puck, the Khepera III posses a powerful hardware able to move by itself more than 1kg just by pushing. Moreover, its maximum speed is 6 $rads/s$, which is equivalent to 11.5 Km/h approximately. This made our implementation improve in 32% the transportation time of a circular object in an open environment.

This means that, by changing the hardware we can enhance the performance of the traditional OBP method time wise. If we want to increase its efficiency, the resting algorithm should be implemented as well as testing the method with real robots to have more accurate results from the swarm. A 3D simulator such as VREP can simulate a swarm, but the noise and the lack of accuracy in certain points of the simulation time makes it a tool that can be replaced by a more powerful one. However, it is easy to use and do not need high performance computers to be installed and used.

6.2 Self-Guided OBP in Unicursal Environments

The OBP method only works as a transportation method for our research. This helped us to focus on developing a method to guide the pusher robots through environments with obstacles. The obstacles are placed as walls that block the direct sight of the robots to the object. The aim of this stage is to overcome the limitations of OBP. For the sake of this aim, the robots explore the environment and place themselves as guides, which help the pusher robots to know where to move the object. Thus, the robot team can retrieve back the object to the Home position.

In the exploration part of the experiment, the robots interact with the environment and other agents in order to visit all the sections of the experiment area. Normally, this process pass without any problem; but as we explained in the previous section, the simulator miss a sensor reading or add too much noise in the simulation. Thus, this make the robots collide with each other and flip themselves to a nonoperative state.

In the experiments 2.a, 2.b and 2.c, this phenomenon is not so appreciable due to the amount of robots interacting with each other. In other words, the simulator is able to handle the amount of robots in this process without any problem. However, in the experiments of experiment 3, we can see robots colliding and flipping each other making the guiding process inefficient. Also, if a robot is close to Home or another guide, makes impossible to a potential guide to be placed. This is due to an important condition of the Guide Mode. Such condition makes the potential guide robot to abandon its duty if another robot is perceived close to another guide or Home. Nonetheless, the robots always aim to finish the task. Therefore, the guides find other places to stay, leading the rest of the robots back to Home.

The experiments have demonstrate that, the amount of robots and the difficulty of the environment increase significantly the time of the process. As mentioned before, if a robot is nonoperative close to a guide or Home, the new guides can not be settle. Therefore, the simulation should be restarted. However, the robots always find the way to retrieve the object if the guides are completely placed. Thus, no matter the conditions of the experiment area the swarm of robots, cooperatively, will complete the task successfully.

6.3 Self-Guided Mode in Maze-Type Environments

As an additional experiment, we created the environment of experiment 4 as a maze-type. The aim of this research was to create an autonomous guiding method for OBP to overcome its limitations. Therefore, unicursal environments would demonstrate enough that our method works as a simple self-Guided algorithm. Nonetheless, we wanted to test the limits of the proposed algorithm to see if its possible to optimize a path in an environment with more than one possible route.

The limitations of our method were explained in the previous section, regarding the flipped robots close to a placed guide or Home. Therefore, in experiment 4, the limitations were not different and a detailed explanation of them are shown in the beginning of section 5.2.4.

Nevertheless, after repeating the simulation, we determined that the swarm of robots was not only able to optimize the path but also, being able to reorganize the path if a mistake is made. As explained in the last experiment, the robots could not see the next guide place in the previous configuration. Thus they pushed the object up to the next wall, changing the configuration of the previous path to take. After the robots perceived this situation, they reconfigured the path placing two more guides in between the section that was far from the fourth guide.

The last configuration of the guide robots, led the pusher robots to do a partial retrieval; not because of the guides configuration, but due to the long time that the simulation took. The robots got stuck whether push the object towards a wall and the rest pushing another wall. However, focusing on the performance of the self-Guiding method; unexpectedly, it gave us a result attributed to the theory and characteristics that defines a decentralized swarm of robots which create intelligence out of a cooperative behavior overcoming the individual limitations.

As a general improvement for the method and implementation we propose:

- shorten the camera range. This will make the robot behind the guides or Home. in order to perceive the guide or home before detecting the other robot behind.

- In the OBP method, implement the resting algorithm. Having this algorithm, the robots will be able to determine when they can arrive to the object and help pushing, without interfering with the rest of the team.

- Use real physical robots for the experiments in order to avoid the noise introduced by the simulation. It is well known that a real environment has more noise than a simulation, but the robot's hardware has the necessary tool to filter such noise.

Moreover, the robot as an unity will take care of their own sample time, in contrast with a simulator that has to sample all the information of all the robots at the same time. This produce several problems regarding the collisions between the robots.

,

Chapter 7

Conclusion and Future work

This thesis introduced a guiding algorithm that work alongside with the OBP method in order to transport objects within unknown environments. Our method makes the group of robots to place themselves in the environment as guides between the object and the destination of the object. After the guiding process finishes, placing the last visible sub-goal, the rest of the robots start the pushing process through the path created by the guides. We have proved that in a simple or complex environment the robots are able to place themselves as guides, leading to a successful retrieval of the object.

According to the experiments and the results obtained we have demonstrated that a swarm of robots with simple instructions and low cost hardware and software can, cooperatively, retrieve an object through complex environments with the help of guide robots leading the team direction without the need of:

- A group leader.
- Human intervention.
- Communication between the robots.
- Separating the robots by a guide team and pusher team.

Nonetheless, in larger environments (such as in Experiment 3 in section 5.2.3 and Experiments 4 in section 5.2.4), the complete self-Guide OBP process takes a significantly long time. However, the object search and retrieval task are accomplished.

The main advantage of our guiding method is that it can be applied to any decentralized system with low-cost robots. Moreover, the robots do not communicate with each other. These characteristics can be taken as limitation, but our system could guide the pusher robots forming a trail between the object position and home. Moreover, the swarm have shown an impressive behavior in reorganizing themselves after a mistake was made by the team. Of course, the algorithm can be improve in order to make the retrieval process faster and efficient, but itself our method can accomplish the task overcoming the limitation of previous works.

To the best of our knowledge, this guiding method is the first successful attempt to retrieve an object based on a swarm using our guiding method, which involves simple steps to take in order to come up with a path that the pusher robots can follow.

Based on the insights we obtained from the results of the experiments, we list up several future directions of this work.

- The resting algorithm is going to be implemented. We expect that by applying this functionality the robots will retrieve the object much faster. Moreover, the aim of this functionality is to overcome the sensitiveness of our method to the number of robots deployed, in order to make a scalable swarm in the future making sure that the majority of the robots will be able to return to Home after the task is completed.

- Extend the Guide Mode of the Guide-Based OPB algorithm in order to let the robots optimize a path from the object back to Home. The optimization will be within a complex environment with spaces that will not lead the robots return to Home, such a dead end of a maze. To observe an example of this type of environment, Figure 7.1 shows an on going experiment we have been developing for this aim.

- Using the Guide Mode algorithm to be implemented in 3D environments. We have found the need of testing our method in places with different heights or zones where the robots can access by stairs or ramps. For these aim, the robots should

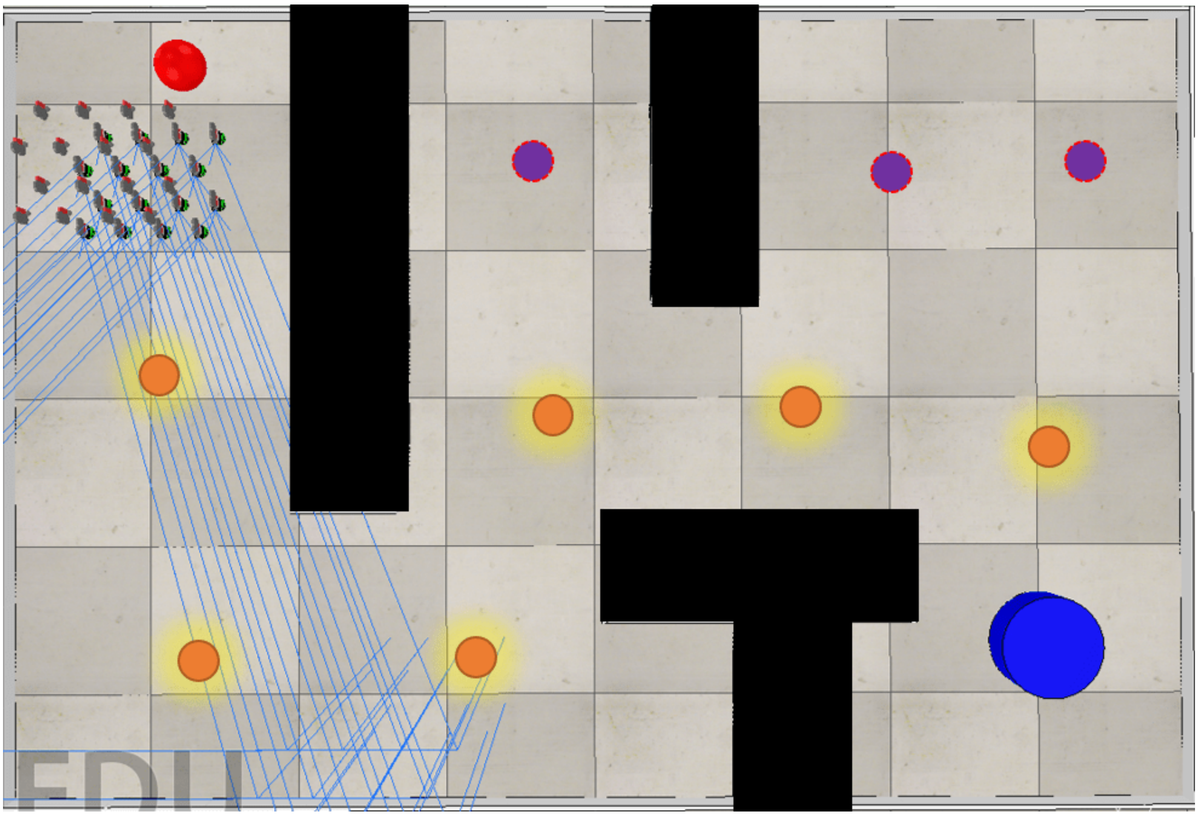


Figure 7.1: Representation of the goals to achieve in the future. The glowing orange dots are the expected positions of the guides after they optimized the path on a complex environment with spaces that will not lead to the red mark (home). The purple dots represent the places where the guides are expected to not place themselves.

be modified in order to sort such terrains, but it might open another research area for mini-roboticists.

Bibliography

[Alk+15] M. H. M. Alkilabi, C. Lu, and E. Tuci, "Cooperative object transport using evolutionary swarm robotics methods," *Proceedings of the European Conference on Artificial Life*, pp. 464–471, 2015.

[Awa+18] F. Awad, M. Naserllah, A. Omar, A. Abu-Hantash, and A. Al-Taj, "Collaborative indoor access point localization using autonomous mobile robot swarm," *Sensors*, vol. 18, no. 2, p. 407, 2018.

[Bas+08] S. Bashyal and G. K. Venayagamoorthy, "Human swarm interaction for radiation source search and localization," in *2008 IEEE Swarm Intelligence Symposium*, Sep. 2008, pp. 1–8.

[Bon+99] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems.* New York, NY, USA: Oxford University Press, Inc., 1999, ISBN: 0-19-513159-2.

[Che+15] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, Apr. 2015, ISSN: 1552-3098.

[Cor] K.-T. Corporation. (). Khepera iii, [Online]. Available: `https://www.k-team.com/`.

[EPF] EPFL. (). E-puck robot, [Online]. Available: `http://www.e-puck.org/`.

[Hab+18] G. Habibi, S. P. Fekete, Z. Kingston, and J. McLurkin, "Distributed object characterization with local sensing by a multi-robot system," in *Distributed Autonomous Robotic Systems*, Springer, Mar. 2018, pp. 205–218.

[Iba13] H. Iba, *Agent-based modeling and simulation with Swarm.* Chapman and Hall/CRC, Nov. 2013, ISBN: 9781138033702-CAT#K31542.

[Ing+17] J. E. Inglett and E. J. Rodríguez-Seda, "Object transportation by cooperative robots," in *SoutheastCon 2017*, Mar. 2017, pp. 1–6.

[Isl+17] K. T. Islam, G. Mujtaba, R. G. Raj, and H. F. Nweke, "Handwritten digits recognition with artificial neural network," in *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*, Sep. 2017, pp. 1–4.

[Lan] Y. Landaez. (). Experiment videos, [Online]. Available: `https://drive.google.com/open?id=1aEDDgPkBsLXYmqVWPA-MfB7mHENb2CjK`.

[Luv+18] D. Luviano-Cruz, F. Garcia-Luna, L. Pérez-Domínguez, and S. K. Gadi, "Multi-agent reinforcement learning using linear fuzzy model applied to cooperative mobile robots," *Symmetry*, vol. 10, no. 10, 2018, ISSN: 2073-8994.

[Mec05] T. Mechanics, "Wheel diameter / distance traveled," Tech. Rep., 2005.

[Mel+15] R. S. Melo, D. G. Macharet, and M. F. M. Campos, "Multi-object transportation using a mobile robot," in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, Oct. 2015, pp. 234–239.

[Pfe+01] R. Pfeifer and C. Scheier, "Understanding intelligence," in. MIT press, 2001, ch. 6.

[Rob] C. Robotics. (). Vrep, [Online]. Available: `http://www.coppeliarobotics.com/`.

[Sad+16] A. K. Sadhu, P. Rakshit, and A. Konar, "A modified imperialist competitive algorithm for multi-robot stick-carrying application," *Robot. Auton. Syst.*, vol. 76, no. C, pp. 15–35, Feb. 2016, ISSN: 0921-8890.

[Sch+08] L. Schmidt, B. Buch, B. Burger, S.-H. Chang, J. Otto, and U. Seifert, *Khepera iii mobile robot practical aspects*, 2008.

[Sha+12] M. Shayestegan and M. H. Marhaban, “A braitenberg approach to mobile robot navigation in unknown environments,” in *Trends in Intelligent Robotics, Automation, and Manufacturing*, S. G. Ponnambalam, J. Parkkinen, and K. C. Ramanathan, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 75–93, ISBN: 978-3-642-35197-6.

[SIC] SICK. (). Efficient solutions for material transport vehicles in factory and logistics automation. commercial magazine, [Online]. Available: `https://www.sick.com/`.

[Tor15] S. Torabi, “Collective transportation of objects by a swarm of robots,” Chalmers University of Technology, Tech. Rep., 2015.

[Wan+05] Y. Wang and C. W. de Silva, “An object transportation system with multiple robots and machine learning,” in *Proceedings of the 2005, American Control Conference, 2005.*, Jun. 2005, 1371–1376 vol. 2.

[Wer91] E. B. Werkowitz, “A computer simulation of braitenberg vehicles,” AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING, Tech. Rep., 1991.

[Wik17] Wikipedia contributors, *Trail pheromone — Wikipedia, the free encyclopedia*, `https://en.wikipedia.org/w/index.php?title=Trail_pheromone&oldid=795196661`, [Online; accessed 21-January-2019], 2017.

[Wik18] ——, *Maze — Wikipedia, the free encyclopedia*, [Online; accessed 21-January-2019], 2018. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Maze&oldid=872989189`.

[Yam+00] A. Yamashita, M. Fukuchi, J. Ota, T. Arai, and H. Asama, “Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3d environment,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 4, Apr. 2000, 3144–3151 vol.4.

List of Publications

1. Ysmaldo Landaez, Hiroshi Dohi and Hitoshi Iba, "Swarm Intelligence for Object Retrieval Applying Cooperative Transportation in Unknown Environments," In ACM Conference proceedings for the 4th International Conference on Robotics and Artificial Intelligence (ICRAI), China, November. 2018

2. Ysmaldo Landaez, Hiroshi Dohi and Hitoshi Iba, "Guide-Based Cooperative Transportation in Unknown Environments," In proceedings of the Evolutionary Computation Society, Evolutionary Computation Symposium, Japan, pp.51–58, December. 2018