Distributed Multi-robot Collaboration for Tasks Allocation based on Optimized Greedy Algorithm Approach

Abderrahmane TAMALI (abdou.t10@gmail.com)
University Ferhat Abbas of Setif
Nourredine AMARDJIA
University Ferhat Abbas of Setif
Mohammed TAMALI
TAHRI Mohamed Bechar University

Research Article

Keywords: Cluster, Multi-Robot, Hostile Site, Collaboration, ROS, SLAM, Swarm-robot, 3D Digitization

Posted Date: October 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2133290/v1

Distributed Multi-robot Collaboration for Tasks Allocation based on Optimized Greedy Algorithm Approach

TAMALI Abderrahmane[1*], AMARDJIA Nourredine[1†] and TAMALI Mohammed[2†]

[1*]Dept. Of Technology, FERHAT Abbes University, LIS Laboratory, Setif, 19000, Algeria.
[2]Dept. Of Electrical Engineering, TAHRI Mohamed Bechar University, ENERGARID Laboratory, Bechar, 08000, Algeria.

*Corresponding author(s). E-mail(s): abdou.t10@gmail.com; Contributing authors: amardjia_nour@yahoo.fr; tamali.mohammed@univ-bechar.dz;
†These authors contributed equally to this work.

Abstract

During the last decades, the intervention using robots in sensitive areas reached appreciable mathematical confidence. Robots are equipped with adequate payload and embed processes using high-performance algorithms oriented topology, statistical observations, ontology, or bio-inspired. These algorithms improve considerably the processing capacity in time savings and computational efficiency. The modified GREEDY approach adopted in this contribution aims to optimize the gain in time and cost of processing for task allocation among a cluster of micro-robots with adequate means for the purpose of identifying sensitive areas. Evaluation of the efficiency of the task's planning process to order each agent micro-robot, we optimally evaluate the cost function by grouping the dependencies; radio connectivity, energy at disposal, and the absolute and relative availability of the agent for itself and within the group. One of the first concerns is to validate the positive trend of the growing number of agents forming the cluster. For this objective, our approach introduces a cluster of three micro-robots. The proposed idea is qualified as an adaptive approach for a mission to identify victims at risk in a challenging environment. Each micro-robot in the cluster is configured to maintain interoperability and collaboration that gain support to evolve in the target scene in order to perform the assigned task. Collaboration algorithms are implemented as an adaptive strategy where it is necessary to optimize agents' mobility according to criteria depending on the characteristic of the place to be identified.

Keywords: Cluster, Multi-Robot, Hostile Site, Collaboration, ROS, SLAM, Swarm-robot, 3D Digitization

1 Introduction

The intervention of human operators after or during events of risk within environments qualified as hostile, such as forests in fire, damaged ruins, sensitive places affected by a natural hazard, potentially and biologically infected/radioactive areas, or even big industrial installations, end generally by unavoidable losses in human, money, and equipment. Substituting the operators involved with robotic machines, equipped with specific means and qualifications can massively reduce losses, especially in human life. At the stage of this observation, research in the field has led to the development of means and methods of action without direct physical human intervention. Technically speaking, we are referring to online operation or robotics with the ability to collaborate if needed and ready to intervene in hostile environments Collaboration in the case of multi-machines is one of the most studied topics in the field of collaborative robotics. The number of robots per operation is a major feature. Admittedly, the smaller the latter ($\simeq 1$ alone) the slower the time of the intervention, the opposite case goes through values more or equal to three (≥ 3 - Case called Cluster or multi-robot for a collection of three machines) to move towards intervention by swarm formation (Swarm-robot). At this stage, the robots must evolve according to a certain harmony deduced from a collaboration plan, we highlight this ability to exchange vital information harmoniously. This collaboration is described by intelligent algorithms [1] which take advantage of many characteristics to minimize losses in information, inter-connectivity and dissipated energy and maximize the effectiveness of the intervention while effectively responding to a predefined task (target search, identification, rescue with recognition). Added to this demand is the search for a better multi-operator control and command strategy [2] which can be Centralized (Fully supervised) or Decentralized (with a certain autonomy). The purpose of this work is to take advantage of advances in the field of cobotics and to proceed with the assembly of a cluster of micro-robots for the supervised and/or intelligent collection of data concerning 3D digitization on so-called aggressive and hostile areas (natural, ruin or industrial).

At the advanced stage of our project, our goals are to carry out background simulations, using three configurations of one, two and then three robotic units which evolve, by collaborating intelligently, in a hostile typed environment in

order to identify a target. visually recognizable.
We note that the importance of the subject lies in its novelty [3, 4] as university research and its qualification for the interface of the outcome of research in the laboratories of the Algerian University towards professional applications for assistance and logistical support to workers in high-risk sites.
The main task of micro-cobots will be collaboration for better distribution and execution of roles and the transmission of relevant information, and collected data [1, 2, 5]. The transmission uses the means and connection strategies (radio) available for the cluster of micro-cobots.
For design purposes, we have adopted the Open-Source ROS solution (Robotics Operating System - with its various modules). ROS [6] will facilitate the adoption of conceptual and physical models to project into reality. The concrete example in this domain is developed by [7].
Moving faster with greater reliability is the first demand in these cases. This situation is very common in cases of fire, gas leak, or following seismic disasters where an individual is unable to move beyond a barrier (a concrete block, etc.). In this type of situation, it is wise to note that the important thing is to secure the lives of the operators who intervene first.
In operational research and for cases of NP problems, we encounter situations with one function with many constraints. In the most dreaded cases, we have to deal with a system of objective functions decorated by multiple constraints. In such conditions, we resort to differentials methods, data statistical validation, or those based on AI. Our case is complex in its reality, we content ourselves with increasing the degree of complexity, as the solutions converge. The basic criteria taken into consideration are:

- Morphology (dimensions, DOF)
- Connectivity (Means, persistence)
- Energy autonomy (energy security)
- The payload (equipment for acquiring useful information)

This data is vital to each unit in the cluster. Each micro-robot evolves following a treatment deduced from the methods of collaboration. We are interested in the relevance of the execution of the task with a minimum of effort and in the shortest possible time.

2 Related Works

The main concern for [8] proposes is to optimize the generated map by reducing errors occurring at the estimation level which is depending on robot ability and the mounted payload. For [9], task coordination for a best multi-robot evolution is to perform a task-based OAP Where [10] focuses effort on developing a metric used to estimate fault level within a swarm of robots. [11] proposes a model based on data correlation (namely; the Correlated Random Walk Model) to efficiently approximate task searching time for distributions of multi-robot systems in large arenas. In recent literature, [12, 13] talk about

bio-inspired techniques to achieve collaboration and sharing state information between a group of pursuing agents vs a group of fast evaders.
In the literature studied, it is clear that collaboration presupposes the confirmation of a certain number of criteria. The information on the state of each agent, the onboard means that each agent can put in adequacy, and finally the persistence of the resources on board allows each one to finalize the requested task.
The case of the hostile site presents a complete specificity, which requires more effort to undertake strategies whose mathematical complexity [14, 15] depends on the criteria imposed by the characteristics of the ecosystem (multi-robot, tasks, ROI, and resources). In situations where the site is hostile, the procedure to follow is much more difficult since it depends little on secondary capacities such as the search for a simple trajectory. the intentions consider the optimal solutions [16] of the routes to be followed in the shortest possible time in order to reach the target to be rescued [17]. For industrial sites, the question completely changes form and compactness. We seek to complete the surveillance in the most relevant way in terms of locating incidents [18].
Other research focuses their efforts on the capacities that micro-robots must have in order to acquire cognitive capacities allowing them to evolve on the site to be studied and this by using reinforced learning methods [19].
Speaking of morphology, Cheetah 3 [20] from MIT or ANYmal from ANYBOTICS, developed at ETH Zurich [21], are two concrete examples meeting DOF requirements in difficult sites with increased aggressiveness.

3 Concept and Model

For the modeling, we took over, within the framework of the simulation, the JETBOT prototype from NVIDIA. well-known micro-robot by the completeness of the equipment mounted as payload (JETSON Nano Micro-controller, LiDAR, Camera, Motorized wheels + Driver, Radio system, and OS/ROS). The URDF (SDF) model has been updated to fit our case study. At this stage, the following elements must be qualified to set up a simulation scene compatible with the criteria and positioning of our approach within the following hypotheses:

- The place (the space of an apartment with spatial complexity) in 3D [6, 10, 18].
- The micro-robots cluster (in three similar machines [22]).
- Possibility of the heterogeneous case [23–25].
- the target (3D image/model of a human - complexity of the target's behavior).

Standard packaging:

- Energy autonomy/payload
- Link stability (Radio connectivity with the cluster)
- Collaborative skills for better performance

- Embedded AI
- Minimum size

The assumptions, in our case, are:

- The cluster is designed with three micro-robots $\mu R_i \mid_{i\in[1,R]}$ on motorized wheels (Driver, Differential) for locomotion.
- Each micro-robot unit is equipped with radio means for transmitting data and maintaining communication links.
- The battery mounted on each robot gives it sufficient autonomy for its activity in the cluster.
- Each robot has an embedded AI algorithmic base allowing it to make decisions for individual or collaborative cases.
- Each unit in the cluster is equipped with the necessary vital equipment (LiDAR, Camera, Motor driver, Sensors, Actuators).
- The cluster of micro-robots is controlled in two modes (Supervised/Collaboration).

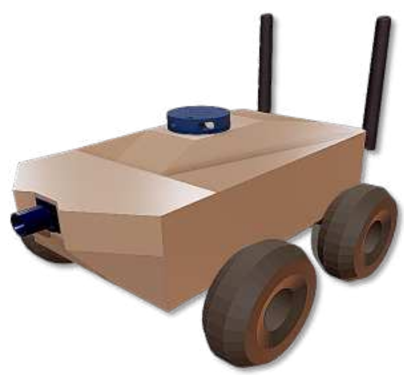


Fig. 1: AL Mustaksheef3D, wheeled robot developed

Our objective is to satisfy the conditions of reliability of the search for victims in a so-called hostile site. One of the material recommendations remains the 3D digitization of the place treated and this follows the fact that we use a LiDAR or a depth camera.
For this situation, the status of success for the detection of victims as well as the time taken to detect it is among the variables to be satisfied. A cost function is defined as an objective function to model the studied case. This function depends on the criteria already mentioned, we expect to minimize our objective function with respect to a set of software and hardware constraints and therefore maximize the success and effectiveness of victim search.
For this purpose, ROS, a Robotics Operating System is qualified to support logic management mounted on the cluster of micro-robots. ROS uses

strategy-based logic nodes or deployed services to ensure the best communication between robotic equipment. Basically, a node can be described as a TALKER or LISTENER, where a talker sends a message and a listener hears the message. The whole treatment in the robotics manner sent or received, depends on the flow of the message. Drivers can then convert the message received by a sender to activate a movement or a robotic action. A ROS service is a process running in the background and listening to client requests. Once a request is detected, an equivalent response can then be sent according to a given set of parameters. Working with ROS is fully documented and

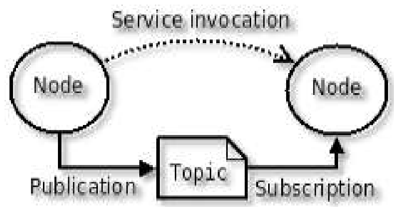


Fig. 2: ROS basics and concepts

where a community is Open to all advice. ROS is an Open Source environment with a complete list of modules and packages. ROS offers great tools for simulation and visualization tasks (GAZEBO, RVIZ, RQT) and also for bridging managed robots within two modes, virtual and real mode. Furthermore and within the interoperability, ROS can interact with a bunch of platforms dedicated to simulating robotics problems [26–29] (such as GAZEBO, Webots, CoppeliaSim, RobotDK, RokiSim, Unreal Engine, Matlab, ABB RoboStudio, NVIDIA Isaac, AirSim and Argos [30]).

The advantages of ROS reside in the capability to be used in programmatic mode by writing codes, generally in C/C++/Python. This method allows users more fidelity and effectiveness of control over robots within their soft and hard components.

This task was successfully executed on the GAZEBO robot simulator with - one robot (effective and relevant result) - three robots (result with a significant level to be optimized through collaboration). The improved version of the ROS-MOVE BASE (ROS Package, responsible for planning robot movement) module is used for collaboration and definition of individual tasks with topological optimization of paths. For the exchange of messages between micro-robots, ROS.MSG (Base ROS message exchanged between ROS nodes and packages) is used for its qualities of simplicity and efficiency.

The IMU data is converted into ODOM to move the robot on its space after validation and voting on the unoccupied area. Collaboration allows in unsupervised mode to manage conflicts on mobility orders on the total space and improve the exchange of vital information.

3.1 Decentralized agent multi-node task allocation

It is difficult to achieve robots acting as cohesive units while being able to distribute tasks to be carried out in real-time. Supervising and coordinating this heterogeneous system requires a decentralized framework that incorporates high-level task scheduling, low-level motion control, and robust, real-time robot awareness [31].

Task location according to multi-robot network architecture is very essential. For the issue of multi-node collaboration, decision and communication are major criteria for task distribution. Communication is a key issue when dealing with multi-robot systems. There is always a strong need for communication with and between robots and even with the control station if it exists.

In self-organized decentralized approaches, each robot node makes its own decisions without major consideration of other agents [32]. They include methods based on or inspired by nature or reality [33] (Swarm Intelligence, Market Strategy, Ant Colony, Distributed Bees), which makes it possible to obtain complex collective behavior from local interactions (Many individuals with simple behavior). In these approaches, sensors use local knowledge and share information with each other [7, 34]. In such systems, sensors work together to achieve an overall goal.

Each robot $\mu R_i \mid_{i \in [1,R]}$ must know the set of tasks to be done. By interacting

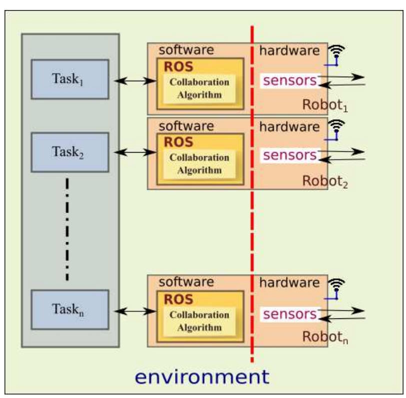


Fig. 3: Cluster architecture

with the place by movement with actuation and by capturing the necessary quantities using well-chosen sensors according to the type of application, the individual decision of each agent requires a specific piece of software processing relevant to each robot.

We adopt Wireless communications to avoid obstacles to agent movement. Therefore, the robustness of the communications in terms of bandwidth, range, power consumption or transmission rate turns out to be a crucial aspect when evaluating the overall performance of the system. The simplest form of communication is a point-to-point scheme, where agents send information directly to its receiver. The choice of means of transmission requires a study of the medium of communication according to the type of information to be transmitted.

3.2 Problem Formulation

Robot evolution on hostile sites can, in some manner, be assumed to be a progression of the vehicle forming a path on a surface contoured by a set of N points in a space defined in a plane delimited by a closed polygon Where

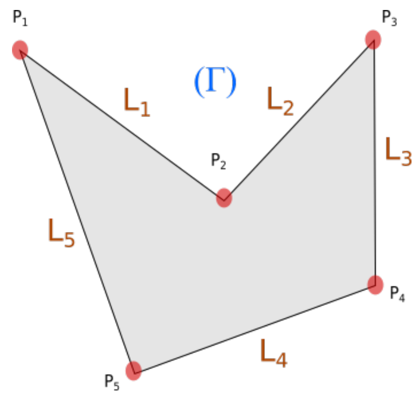


Fig. 4: N-gon of a scanned area

$P = P_1, P_2, ..., P_N$ define a Poly-point or a set of N points (Fig.4); Normally each point is located on the plan (Γ) by its Cartesian coordinates x_i and y_i. A one-line equation can be written in the form:

$$y = a.x + b \tag{1}$$

a and b are two parameters related to the j^{th} line (slope and y-intercept) in the Polyline defined by P components. By using each couple of points coordinates, the related a and b parameters are obtained according to:

$$a = \frac{y_2 - y_1}{x_2 - x_1} \tag{2}$$

$$b = \frac{x_2.y_1 - x_1.y_2}{x_2 - x_1} \tag{3}$$

using Cramer's rule.
so, (Equ.1) can be like follows for a line equivalent equation (L_1) between two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$.
y is given by:

$$= \begin{cases} \frac{y_2-y_1}{x_2-x_1} + \frac{x_2.y_1-x_1.y_2}{x_2-x_1} \mid_{\begin{array}{c} x_1 \le x < x_2, \forall x_1 - x_2 < 0 \\ x_2 \le x < x_1, \forall x_1 - x_2 > 0 \\ y_1 \le y < y_2, \forall y_1 - y_2 < 0 \\ y_2 \le y < y_1, \forall y_1 - y_2 > 0 \end{array}} \\ 0 \; otherwise \end{cases} \tag{4}$$

now, we need to construct a Polygon equation using a combination of multiple line equations. We have N points, which implies that the number of line equations is N $(L_1, L_2, .., L_N)$.
the n-gon's (polygon) formula is given by:

$$Y = Y_1 + Y_2 + ... + Y_{N-1} + Y_N = \sum_{a}^{N-1} Y_i + Y_N \tag{5}$$

witch is $Y_1, Y_2, ..., Y_N$ or simply Y_i,Y_N where $i \in \mathbb{N}$ natural strictly positive number, the set $Y - i$ with $i \in \mathbb{N}$ represent each i^{th} line's equation

$$= \begin{cases} \frac{y_{i+1}-y_i}{x_{i+1}-x_i} + \frac{x_{i+1}.y_i-x_i.y_{i+1}}{x_{i+1}-x_i} \mid_{\begin{array}{c} x_i \le x < x_{i+1}, \forall x_i - x_{i+1} < 0 \\ x_{i+1} \le x < x_i, \forall x_i - x_{i+1} > 0 \\ y_i \le y < y_{i+1}, \forall y_i - y_{i+1} < 0 \\ y_{i+1} \le y < y_i, \forall y_i - y_{i+1} > 0 \end{array}} \\ 0 \; otherwise \end{cases} \tag{6}$$

Equation valid for $1 \ge i \ge N - 1$, and the last line (L_N) :

$$= \begin{cases} \frac{y_1-y_N}{x_N-x_1} + \frac{x_1.y_N-x_N.y_1}{x_1-x_N} \mid_{\begin{array}{c} x_1 \le x < x_N, \forall x_1 - x_N < 0 \\ x_N \le x < x_1, \forall x_1 - x_N > 0 \\ y_1 \le y < y_N, \forall y_1 - y_N < 0 \\ y_N \le y < y_1, \forall y_1 - y_N > 0 \end{array}} \\ 0 \; otherwise \end{cases} \tag{7}$$

for $a, b, \alpha \in \mathbb{R}$ and $f(x) = y = \alpha$. The area inside the irregular polygon can be defined as the result of:

$$Area_Y = \int_{\mathbb{R}} \alpha.d\alpha \mid_{\begin{array}{c} f(x) = \alpha = Y = \{x';\ x'';\ \ldots;\ x^n;\ x^{n+1}\} \\ where \, x^n,\ x^{n+1} \in \gamma(x) \end{array}} \tag{8}$$

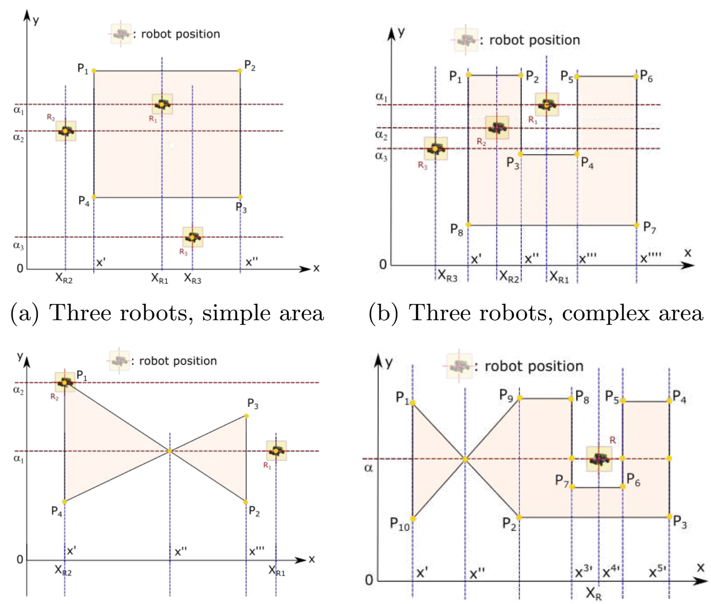
Such as $n = 2m + 1 \mid_{m\in\mathbb{N}}$, $x^n < x^{n+1}$, and $\gamma(x)$ represent the variation domain of variable x.

So, the area viewed by each robot can be mentioned as Area(k) where k=1,2,. . . ,R and R is the number of robots.

Therefore, we can extract from Area(k) equation a relation $g_k(\alpha, x)$ between the variables α and x which verifies the condition of which

$f(x) = \alpha = Y = \{x';\ x'';\ \ldots.;\ x^n;\ x^{n+1}\}$. This behind (g_k) helps us to determine if the robot k is in the surface Area(k) or not.

Regarding, the determination of whether the robot is outside a polygon area or inside. We take four zone-shaped situations including most of the possible cases Figures 5a, 5b, 5c, 5d.



(a) Three robots, simple area (b) Three robots, complex area

(c) singular posture case (d) Insoluble problem case

Fig. 5: Possible cases of robot's posture.

for the first situation (Fig. 5a), robot-1 is located in the zone and the other robots (robot-2 and robot-3) are not in it. To formulate the situation, we have $g_1(\alpha_1, x) = g_2(\alpha_2, x) = \{x', x''\}$ and $g_3(\alpha, x) = \emptyset$.

So, by comparing between x_{R_k}, x' and x'' we can conclude that:
- if $g_k(\alpha, x) = \emptyset \Rightarrow$ the robot k out of zone.
- if $g_k(\alpha, x) \neq \emptyset$ and $x' \leq x_{R_k} \leq x'' \Rightarrow$ the robot k in the zone.
- if $g_k(\alpha, x) \neq \emptyset$ and $(x_{R_k} < x'$ or $x'' < x_{R_k}) \Rightarrow$ the robot k out of zone.

The same for the second (Fig. 5b). We have $g_1(\alpha_1, x) = g_2(\alpha_2, x) = g_3(\alpha_3, x) = \{x', x'', x''', x''''\}$, robot-2 is in the area of which $x' \leq x_{R_2} \leq x''$ and others are not, $x'' < x_{R_2} < x'''$ and $x_{R_2} < x'$ which implies that:
- if $g_k(\alpha, x) \neq \emptyset$ and $(x^n \leq x_{R_k} \leq x^{n+1})\ |_{n=2m+1, \forall m \in \mathbb{N}} \Rightarrow$ the robot k in the area.

- if $g_k(\alpha, x) \neq \emptyset$ and $(x^n < x_{R_k} < x^{n+1}) \mid_{n=2m+2, \forall m \in \mathbb{N}} \Rightarrow$ the robot k out of area.

The other situation (Fig. 5c), it represents two singular specific cases including $g_1(\alpha_1, x) = \{x', x'', x'''\}$ and $g_2(\alpha_2, x) = \{x'\}$. It is possible to know if the robot is in the area only in these two cases where $\mid g_k \mid= 3$ for robot-1 and $\mid g_k \mid= 1$ for robot-2 such as:
- if $g_k(\alpha, x) \neq \emptyset \mid_{|g_k|=3}$ and $x' \leq x_{R_k} \leq x''' \Rightarrow$ the robot k in the area. - if $g_k(\alpha, x) \neq \emptyset \mid_{|g_k|=1}$ and $x' = x_{R_k} \Rightarrow$ the robot k in the area.

The last (Fig. 5d), the case where it is impossible to know using the relation g_k; whether the robot is in the zone or not whose $g_k(\alpha, x) \neq \emptyset$, and $\mid g_k \mid= n \mid_{n=2m+5, \forall m \in \mathbb{N}}$

3.3 Greedy algorithm for decentralized task allocation

A greedy algorithm 1 [33] for a multitasking observation problem with broadcast messaging is presented, this algorithm is configured to perform sensor (agent) allocation based on the best possible allocation of each individual sensor to a task that maximizes the performance-to-cost ratio (V_{ik}/d_{ik}), where V_{ik} is the performance of the k^{th} sensor on the i^{th} task and d_{ik} is the Euclidean distance between the sensor and the task (Equ.1).

$$\mathrm{d}_i^k = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2)} \tag{9}$$

$$\mathrm{T}_i^k = \underset{i \in T}{max} \left(\mathrm{V}_i^k / \mathrm{d}_i^k\right) = \underset{i \in T}{max} \left(\mathrm{V}_i^k . \eta_i^k\right) \, where \, \eta_i^k = 1/\mathrm{d}_i^k \tag{10}$$

where task i is the chosen task of the k^{th} sensor out of all possible tasks for the assignment, and T is the group of tasks in the k^{th} sensor range out of M available tasks:
The greedy algorithm (1) proceeds as follows. First, defining the surface variables explored by each robot k: $S^k \mid_{k \in \mathbb{R}}$ denotes the set of boundary points of the surface currently scanned, and $f_i^k \mid_{k \in \mathbb{R}}$ represents an the objective function (cost function) for each robot k and task i stored in t_i^k.
Moreover, T_i^k is the set of tasks that have not yet been allocated and are enclosed in T the total available tasks $(T_i^k \in T)$ while T^k for $k\mathbb{R}$ is the chosen task for the robot, P_i^k is that of the spots that have already been affected. K, the robot that needs to update its bid at the current stage.
Initially, no task is assigned, so $T^k = \emptyset$ for all $k \in \mathbb{R}$. At each step, exactly one task is allocated for a single robot and in an independent way (the principle of decentralization which forces each robot to make its decisions independently and in coordination with the others), so we need T steps, the number of tasks of the current robot, so that is in a state of completeness. At each iteration i, after removing the conflicting parts with the other areas explored by the other robots, all the robots $k \in \mathbb{R}$ submit an offer, which consists of the pair $(t_i^k, T_i^k$. Each robot k chooses the task T^k from the list of non-located tasks T_i^k, such

Algorithm 1 Greedy Algorithm Based Task Allocation for each $\mu R_{k'}$

Environment Initialization
$S^k \leftarrow \emptyset$
$Ko \leftarrow k'$ ▷ $\forall k' \in 1, 2, ..., R$, R nb of robot
$T^k \leftarrow \emptyset$
get(P_i^k) ▷ get all previous task P_i^k if they exists or put it equal to $\emptyset$
Task allocation Process
for $(k = 1$ to $k = R)$ **do**
    $S^k = Area(k)$
end for
for $(k \leq R)$ **do**
    if $(k \neq Ko)$ **then**
        $S^{Ko} = S^{Ko} - S^{Ko} \cap S^k$
    end if
end for
$T = task(Ko) \mid_{S^{Ko}}$
$i \leftarrow 1$
while $i \leq \left|T\right|$ **do**
    $t_i^{Ko} = f_i^{Ko} \mid_{S^{Ko}}$ ▷ f objective function
    if $(T_i^{Ko}(t_i^{Ko}) \subset P_i^{Ko})$ **then**
        $t_i^{Ko} = \emptyset$
    end if
    $i \leftarrow i + 1$
end while
$T^{Ko} = T_i^{Ko} \mid_{max(t_i^{Ko}), S^{Ko}}$
if $T^{Ko} \neq \emptyset$ **then**
    $P_i^{Ko} = P_i^{Ko} + T^{Ko}$
else if then
    $T = \emptyset$
    for $k = 1$ to $k = R$ **do**
        if $k \neq Ko$ **then**
            $T = T + task(k) \mid_{S^k, k \neq Ko}$
            $i \leftarrow 1$
            while $i \leq \left|T\right|$ **do**
                $t_i^k = f_i^{Ko} \mid_{S^k, k \neq Ko}$
                if $T_i^k(t_i^k) \subset P_i^k$ **then**
                    $t_i^k = \emptyset$
                end if
                $i \leftarrow i + 1$
            end while
        end if
        $T^k = T_i^k \mid_{max(t_i^k), S^k, k \neq Ko}$
    end for
    $T^{Ko} = max(T^k) \mid_{k \in 1,2,...,R}$
    $P_i^k = P_i^k + T^{Ko} \mid_{k \in 1,2,...,R}$
end if

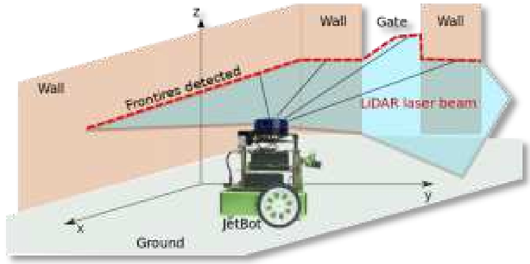


Fig. 6: JetBot during environment identification

that it obtains the best optimal gain with respect to the individual objective function f_i^k. After collecting all the bids, we find that we have a better optimal gain with respect to the collective objective: the multiplicative success of group F (Equ.11). Thanks to the bidding-based formulation, we can efficiently choose the task pair-robot that gives the best collective gain [35, 36].

$$f = max_{\{T^k\}_{k\in\mathbb{R}}} \prod_{k\in R} f_k(T^k) \tag{11}$$

In singular cases where other robots surround the robot, such as the surrounded areas make a false indication at the level of the robot so that it cannot explore further in terms of space for evolution. To ensure performance in cases of encirclement where $T^k = \emptyset$, we, therefore, seek a better offer that maximizes the cost function of the robot by using the available spots of the other robots until the robot is independent.

3.4 Simulation And Success Factors

3.4.1 Adopted OS and Workstation

A personal computer, acts as a collaborative scenario simulator to seek out a victim while exploring. (Tab.1) shows the basic technical specifications of the computer that is used for this test bench. ROS, which is an open-source middleware, was chosen as the software platform to accomplish this simulation. The first and well-supported operating system for ROS is Linux Ubuntu 20.04. ROS is a set of software packages and principles that aim to reduce software

Table 1: Table to test captions and labels.

Product	HP ProBook x360 435 G7
Processor	AMD Ryzen 7 PRO 4750U
GPU card	AMD Radeon RX Vega 7
RAM	32 GB DDR4 Kingston
Storage SSD	1 TB Samsung

complexity and simplify communication between logical nodes and save developers time by supporting code reuse in robotics research and development.
It is designed to be a distributed computing environment, where a number of components such as robots and computers are networked together to communicate with each other by passing messages, using a publisher and subscriber or client and server model [37]. The ROS architecture has been designed and divided into three levels of concepts [37]:

- The file system level: In this level, a group of concepts is used to explain how ROS files are organized on the hard disk. The most basic unit of ROS is ROS packages. They contain one or more programs (nodes), libraries, messages, etc., which are organized together as a single unit
- The computational graph level is the peer-to-peer network of ROS processes that process data together. The main concepts of this level are ROS nodes, master, parameter server, messages, subjects, services and bags. Each node in the system can access this network and communicate with other nodes using messages that are transported using named buses called subjects. The ROS master provides naming and registration services to nodes in the ROS system. It tracks editors and subscribers to topics. Without the master, the nodes could not find each other and exchange messages.
- The community level: which includes a set of tools and concepts to share knowledge, algorithms and code between developers.

3.4.2 Mobile robot model

The robot used in our simulation meets certain criteria [29]. it has a small size

Fig. 7: JetBot from NVIDIA

compared to the limits of the environment, can communicate over Wireless with a computer and/or other robots, has an open software and hardware development model source, and is available on a suitable number of platforms (NVIDIA (source https://github.com/NVIDIA-AI-IOT/jetbot)).
To take into account the above points, the mobile robot JetBot (Fig.7) is

used in the simulation. JetBot (as defined on NVIDIA's official site) is an open-source AI robotics platform that gives builders everything they need to make creative AI applications, intelligent and captivating. It's based on the compact and powerful NVIDIA Jetson Nano computer for AI, which supports multiple sensors and neural networks in parallel for object detection, collision avoidance, and more. This highly innovative robotics platform offers a wide variety of configurations allowing users to implement custom applications.
JetBot is a differential mobile robot with two stepper motorized wheels. With a length and width of 315 mm and 210 mm respectively. Basically, the height depends on the extensions connected. Its hardware and software are fully open source. It has a radio link that can be chosen from WiFi, Bluetooth, ZigBee, and others to connect to computers or to communicate with other robots, and it can work in a swarm formation. Using three JetBots, a team of robots communicating with each other is formed, to be used in the validation of the distributed collaboration algorithm.

3.4.3 Our Approach Based Algorithm

The overall process is illustrated in the next flowchart (Fig.8) Once tasks are

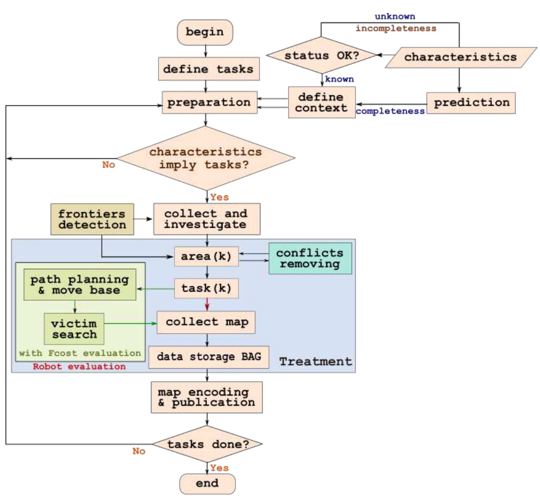


Fig. 8: Flowchart of our approach

defined, the cluster aims to honor the request in more than one step. Preparation: cluster must collect necessary data related to the context of the ROI. We have two responses to the question; the characteristics of the context are almost known and complete, so it's advisable to go next step of the remaining processing.
If not, we are facing an incompleteness case where a prediction strategy is used

to fulfill the data requirements before beginning the next process. If characteristics imply requested tasks, an investigation is launched to detect frontiers surrounding each micro-robot and then collect a piece of the necessary information about its location. This information is needed for voting the segmentation of all scanned areas as specific tasks assigned to each unit of the cluster (path planning and victim search). In this step, fragments of the context's map are generated while the related data is written on a storage mean.
After the previous step, the map's fragments are collated and encoded in a special format before being shared within the cluster. At this stage and if all tasks are done, processes can dispose of otherwise, we jump to the preparation step and redo the following treatment.
In the real case, each unit (micro-robot) must satisfy the optimal condition established by the cost function f_{cost} (Equ.12). The requirements for a real micro-robot in a real terrain, are all these life conditions to be maintained during a mission such as:

- Radio connectivity to the sink (AP for Access Point) or to the cluster $F_{con/AP}$ which guaranty an exchange link with the group.
- Energy autonomy F_{auton}, which guaranty a battery lifetime for a specific mission.
- Absolute Availability $Disp^{All}$, Represents the state OK of the unit within the cluster.
- Relative Availability $Disp^{/Res}$, represents the availability of a relative resource as a payload to be carried on for a specific mission.

$$f_{cost} = k_1 * F_{con/AP} + k_2 * F_{auton} + k_3 * Disp^{All} + k_4 * Disp^{/Res} \quad where\ k_1 + k_2 + k_3 + k_4 = 1 \tag{12}$$

k_i for $i \in [1-4]$ are the respective weight of each part of the agent's function cost. As the best solution for each agent in the cluster, we seek the optimal result (Equ.13) of the function f_{cost}, then:

$$f_{cost}^{opt} = Max\ |_{i\in[1,R]} \left\{ f_{cost}^{(i)}, constraints \right\} \tag{13}$$

represents the optimal value, where the cluster can declare a status OK for the availability and can surpass the minimum required condition to execute tasks assigned.
We consider $F_{con/AP}$ as the reduced effective availability of at least a link with a predefined access point. If We know that F_{con}^{ref} is the reference threshold of a WiFi connection to an AP and $F_{con}(t)$ is an instance connectivity level of an agent to the AP, we define $F_{con/AP}$ as the ratio of instant connectivity $F_{con}(t)$ to the reference threshold F_{con}^{ref} what gives:

$$F_{con/AP} = 100 * (F_{con}(t)/F_{con}^{ref}) \tag{14}$$

in percent (%).
Therefore F_{auton} is the battery autonomy, defined as the remaining energy in Ah of the battery needed to provide adequate power to the agent to perform the assigned task as one unit within the cluster. We define τ the estimated time for the assigned task, I as the battery's actual debited current in Ampere (A), and finally C the battery capacity as a current source for a considered time in Ampere-hour (Ah). Then, is:

$$F_{auton} = C/\tau * (1/I - 10/P_{charge}) \tag{15}$$

For the estimation of $Disp^{All}$ and $Disp^{/Res}$, the availability of a cluster member is defined by its ability to be effectively used or not for any assigned task. Therefore, relative availability with respect to a resource is the agent's ability to have the indicated resource (payload) available at the appropriate time if a task is assigned when the absolute availability is the conjugation of all relative availability. which is equivalent to the fact that all the agent's resources are ready to be used. The relative availability is equal to 1 if there is OK feedback following the interrogation of the resource and 0 otherwise. Then:

$$D_{Rel/Res} = \begin{cases} 1\, if\, the\, targeted\, resource\, is\, available \\ 0\, otherwise \end{cases} \tag{16}$$

While the absolute availability is given by:

$$Disp^{All} = \bigcap_{i=1}^{Nres} (D_{Rel/Res^i}) \tag{17}$$

In other words, it is the logical AND connection of all the relative availability functions which qualifies the certain capacity that the agent can intervene with all its payload in a given task.
The weighting coefficients k_i are chosen in many ways so that the final result of the objective function is optimal. We use bio-inspired methods to determine these coefficients.

4 Simulation results and Discussion

The scene of the simulation scenario is given in (Fig.9). The scene is characterized by more than one constraint to qualify the ability and reliability of our approach in terms of a cluster of micro-robot navigation with or without collaboration.

Mainly, The scene is composed of an area of a house delimited by exterior walls. The interior of the house is divided into rooms where each room contains more or less furniture. To enlighten the situation, we suppose that for some reason more than one victim is located on the scene and needs urgent rescue when a human can't access it. The main request is to share rescue

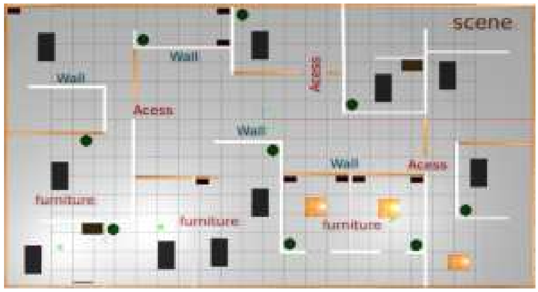


Fig. 9: Context model of the simulation.

recommendations between a cluster of micro-robots.
The scenario was founded on three levels of severity to demonstrate the effect

Fig. 10: Context model with victims' location.

of introducing collaboration and what's influence can have this strategy to reduce effectively the rescue time and increase significantly the reliability of the overall identification task.
From the first to the third level, one single micro-robot then a team of two micro-robots, and finally a small cluster of three units. The simulation (Figures 12,13,14 and 15) was done without and then with the Collaboration strategy. We assume that the simulation run under the following hypothesis:

- The scene domain and dimensions are invariant during investigation time.
- Target position within the scene doesn't have an effective impact on simulation time duration.
- We assume agents (micro-robots) homogeneous (with exactly the same characteristics).
- Connectivity, Payload, and Autonomy conditions are OK for all the cluster units.

Results of the simulation are given in the next tables and figures. The adopted

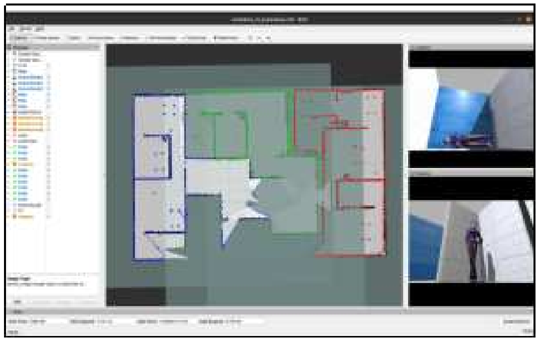


Fig. 11: Space segmentation for three micro-robots, target detection snapshots (RVIZ visualization).

strategy is to calculate the search times made to reach each victim (target), namely that the extreme duration to reach the two victims (given a finite number of targets) represents the maximum of the two operation's times duration. Ten trials (Tab.2) are processed to finally calculate the average of the maxima. We are very interested in the maximum and minimum values of the average duration.

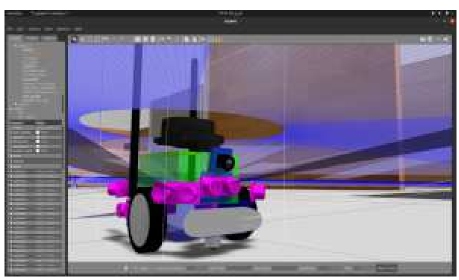


Fig. 12: One Agent's Cluster in action on GAZEBO Simulator environment.

Fig. 13: GAZEBO Simulator showing the scene.

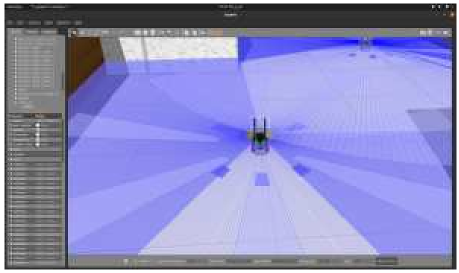


Fig. 14: One Agent's Cluster within a SLAM operation.

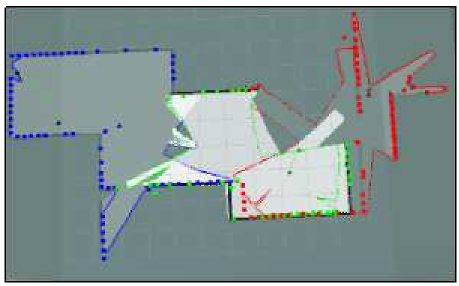


Fig. 15: Agents ROI Updates during SLAM investigation on RVIZ Interface.

The areas illustrated in figure 15 in three colors RGB indicate the allocated zones in the All Area. These three zones are the segmentation of the overall ROI using the Greedy Algorithm (Inference implemented in each agent of the cluster).
The scenario is repeated three times, for the first with one robot measured in ten attempts, results are given by (Tab. 2). Where graphic distribution by

Table 2: Time to reach the target in minutes.

Attempt No	**Victim 1**	**Victim 2**	**til task's end**
1	46	90	90
2	77	112	112
3	95	150	150
4	9	76	76
5	110	123	123
6	55	102	102
7	70	30	70
8	20	150	150
9	93	35	93
10	64	86	86
Average	**63.9**	95.4	105.2

number of attempts is given in the next chart (Fig.16).

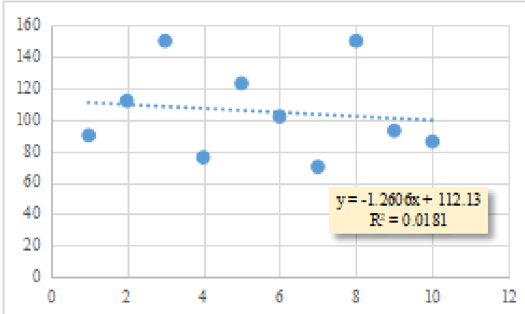


Fig. 16: Time duration to reach the target by one robot for the first experiment.

These procedures of ten simulations are repeated three times each, while the results are considered for five separate cases (Cases I to V) studies. In each trial, we note the time taken to reach all targets in the absence or consideration of the collaboration measure:

- Case I: One robot and two targets (Tab. 3)
- Case II: Two robots and two targets without collaboration (Tab. 4)
- Case III: Two robots and two targets with collaboration (Tab. 5)
- Case IV: Three robots and two targets without collaboration (Tab. 6)
- Case V: Three robots and two targets with collaboration (Tab. 7)

Table 3: One robot average time (min) **OneRobot**

	victim 1	**victim 2**	**max duration**
Experiment 1	63.9	95.4	105.2
Experiment 2	67.5	99.5	120.6
Experiment 3	80.2	37.5	90.7

Table 4: Two robots average time (min) **2NoCollab**

	victim 1	**victim 2**	**max duration**
Experiment1	62.4	40.3	66
Experiment2	72	61.6	95.2
Experiment3	72	38.2	85.3

Cumulative results are collected as a minimum and maximum values for possible consolidation (Tab. 8).

Table 5: Two robots average time (min) **2WCollab**

	victim 1	**victim 2**	**max duration**
Experiment 1	60.1	38.9	62
Experiment 2	33.8	30.4	47.5
Experiment 3	27.2	21.7	34.2

Table 6: Three robots average time (min) **3NoCollab**

	victim 1	**victim 2**	**max duration**
Experiment 1	71.2	29	73.9
Experiment 2	57.8	39.3	69.1
Experiment 3	72.6	32.2	76.5

Table 7: Three robots average time (min) **3WCollab**

	victim 1	**victim 2**	**max duration**
Experiment 1	18.4	14.1	20.7
Experiment 2	18.3	21.5	24
Experiment 3	22.7	7.9	22.7

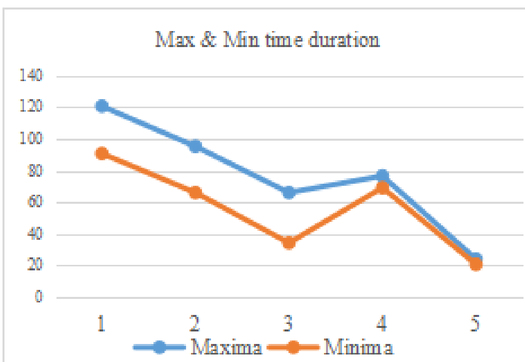


Fig. 18: Max and Min time duration cs experiment.

The collaboration is increasingly affecting the time duration (Fig.17,18). Also, the more the number of agents is more significant the duration time is lower, especially if a collaboration strategy is combined.

The ROI area was processed in two ways, one by locating the wanted targets, and the other by digitizing the location for possible reconnaissance. This last operation was performed by an improved version of a horizontal LiDAR (RPLiDAR A1M8). This composition is one of the novelties introduced on the AL Moustaksheef3D platform, a robotic unit under development.

The new LiDAR has been tested on machines with wheels and on a drone, anchored on the lower base of the latter and oriented downwards.

The following figure 21 gives an overview of the data collected in PCD (Point

Table 8: Time duration average by experiment

	Exper. 1	Exper. 2	Exper. 3
OneRobot	105.2	120.6	90.7
2NoCollab	66	95.2	85.3
2WCollab	66	47.5	34.2
3NoCollab	73.9	69.1	76.5
3WCollab	20.7	24	22.7

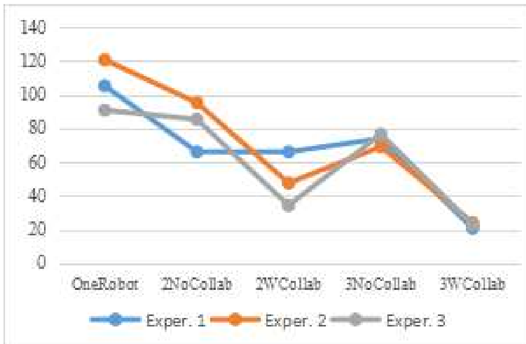


Fig. 17: Evolution of time duration by experiment.

Table 9: Max and Min time duration.

	One	2NoCol	2WCol	3NoCol	3WCol
Maxima	120.6	95.2	66	76.5	24
Minima	90.7	66	34.2	69.1	20.7

Cloud) format from the investigation area. It should be noted that this LiDAR model can be used in two ways in combination. It is the result of an improvement using two LiDARs, one in a horizontal position giving geolocation information in relation to borders and the other providing information on the additional 3D data necessary to complete and form a 3D vision. of the studied place.

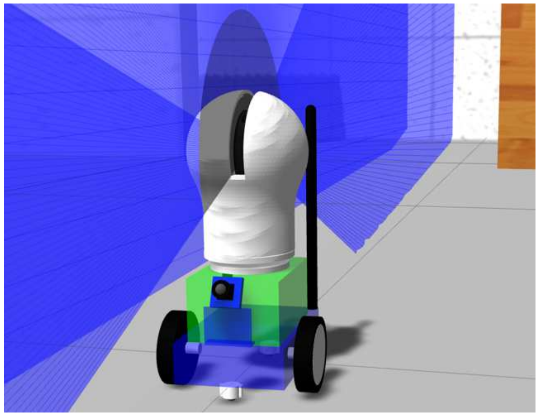


Fig. 19: JetBot with the new LiDAR.

Fig. 20: Perspective view of the new LiDAR.

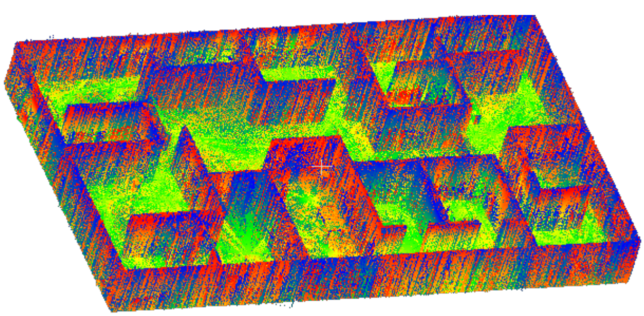


Fig. 21: Point Cloud RGB Restructured.

The reproduction of the scene is the origin of relative data which identification is made for a good estimate of the facts which compose the place to be studied. The collected point-cloud data is used for the photogrammetric reconstruction of the elements of the scene. Tools like CloudCompare, MeshLab, Blender, and even Gimp, as an Open-Source software suite, are capable of imposing background processing (Fig. 22) on point cloud data. This software exists in the API version which proliferates the possibilities of being integrated as an embedded process on each micro-cobot agent of the cluster. The advantage of these treatments based on very powerful algorithms is the aid to the selective identification of targets in the scene.

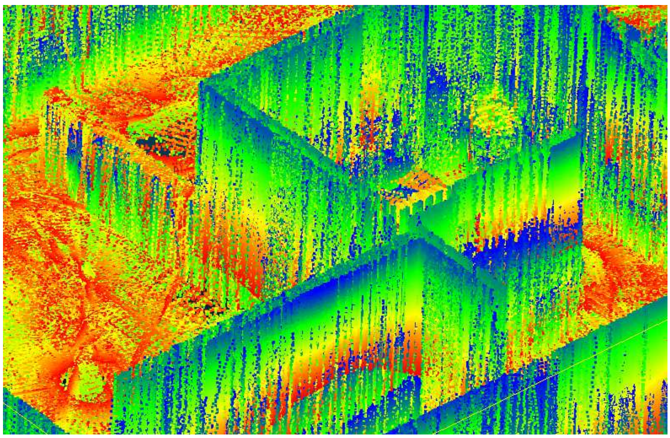


Fig. 22: Point Cloud RGB Restructured.

In terms of consolidation, we have introduced a task allocation approach for the intelligent identification of targets in a hostile site. We have implemented a logical strategy based on the estimation of the function of a cost processed by the improved Greedy algorithm. The influence of the collaboration between the different agents of the cluster proves to be very profitable to reduce in a very interesting way the overall time of the assigned task, however, this strategy requires very expensive capacities from the point of view of the processor. Admittedly, the basic calculation is centralized according to the sense of collaboration and is decentralized if we consider that each agent of the cluster manifests itself independently to judge its decisions, once a task is associated with it.

Conclusion

The focal framework of this work is to highlight this problem and to propose a basic design of a cluster (group) of cobots having the material faculty and endowed with intelligence, to evolve in a hostile environment in order to identify a target and therefore lighten the human task (response or rescue team). These techniques use robots with collaborative capabilities, commonly called

cobots because they coexist safely with human operators. In these places, the spatial occupation density of robots and technicians carrying out routine surveillance poses a major risk to the safety of human lives. The exhaustiveness of the information exchanged risks being lost or incomplete due to this density.
In this sense, our current research focuses on the group of micro-robots in collaboration with humans, hence the notion of micro-cobots, in order to reduce the payload on one side and minimize bottlenecks in other's tasks. In our case studies and according to the results of the simulations with the ROS environment and its visualization and debugging tools, it has been shown that the number of micro-robot agents and the combination of their capacities considerably improves the time of investigation and search for targets. We demonstrated that the investigation space has no influence on the adoption of a cluster of N_R robot agents, even more on the consideration or neglect of a collaboration strategy. The position of the targets did not suggest any modification, once the same tools are used by each search agent.
This result effectively proves our first consideration and this is without forgetting the preestablished hypotheses.
Once more, Charts 17 and 18 show, that collaboration introduces a big improvement and gives the cluster the strength to move forward in the target search operation. If no collaboration is activated within the cluster, micro-robots act strangely and duration time goes high. from the statistical point, the dispersion is significant if the number of agents in the cluster is small. It is expected that this quantity will have a random variation in cases where collaboration is not taken into consideration. It is to be expected, for a scenario without collaboration, a general chaotic state, and the collapse of the system. For this purpose, a reasonable number of investigators in a cluster is preferable ($1 < N_R \leq 3$). For more, the situation is handled with swarm techniques, and collaboration is taken into account.

5 Declarations

5.1 Competing interests

The authors of this work declare that there are no conflicts of interest as defined by Springer, or other interests which could be perceived as influencing the results and/or the discussion reported in this article.

5.2 Authors' contributions

We acknowledge that the results of the experiment, necessary data, and figures in this manuscript have not been published elsewhere, nor are they under consideration (all agree with this) by any other publisher.

5.3 Funding

No funding was required/used to prepare this work

References

[1] Montori, F., Bedogni, L., Bononi, L.: A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring. IEEE Internet of Things Journal (2018). https://doi.org/10.1109/JIOT.2017.2720855

[2] Dong, Y., Zhang, L., Liu, Y.: Experimental study on control strategy of cobot micro operation force hoisting system. Gaojishu Tongxin/Chinese High Technology Letters (2007)

[3] Müller, V.C., Bostrom, N.: Future progress in artificial intelligence. AI Matters (2014). https://doi.org/10.1145/2639475.2639478

[4] Boman, J., Taylor, J., Ngu, A.H.: Flexible iot middleware for integration of things and applications. In: 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 481–488 (2014). https://doi.org/10.4108/icst.collaboratecom.2014.257533

[5] Saied, Y.B., Olivereau, A., Zeghlache, D., Laurent, M.: Lightweight collaborative key establishment scheme for the Internet of Things. Computer Networks (2014). https://doi.org/10.1016/j.comnet.2014.02.001

[6] Wang, X., Ma, X., Li, X., Ma, X., Li, C.: Target-biased informed trees: sampling-based method for optimal motion planning in complex environments. Journal of Computational Design and Engineering **9**(2), 755–771 (2022). https://doi.org/10.1093/jcde/qwac025

[7] Stoelzle, M., Miki, T., Gerdes, L., Azkarate, M., Hutter, M.: Reconstructing Occluded Elevation Information in Terrain Maps With Self-Supervised Learning. IEEE Robotics and Automation Letters **PP**, 1 (2022). https://doi.org/10.1109/LRA.2022.3141662

[8] Rekleitis, I., Dudek, G., Milios, E.: Multi-robot collaboration for robust exploration. Annals of Mathematics and Artificial Intelligence (2001). https://doi.org/10.1023/A:1016636024246

[9] Gerkey, B.P., Mataric, M.J.: Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), vol. 3, pp. 3862–38683 (2003). https://doi.org/10.1109/ROBOT.2003.1242189

[10] Lee, S., Milner, E., Hauert, S.: A Data-Driven Method for Metric Extraction to Detect Faults in Robot Swarms. IEEE Robotics and Automation Letters **PP**, 1 (2022). https://doi.org/10.1109/LRA.2022.3189789

[11] Yi Zhang, D.B.: A Correlated Random Walk Model to Rapidly Approximate Hitting Time Distributions in Multi-Robot Systems. In: IAS-22 (ed.) Https://www.ias-17.org/, pp. 2–12. IAS-22, Zagreb (2022)

[12] Fu, X., Zhang, Y., Zhu, J., Wang, Q.: Bioinspired cooperative control method of a pursuer group vs. a faster evader in a limited area. Applied Intelligence (2022). https://doi.org/10.1007/s10489-022-03892-8

[13] Merkulov, G., Weiss, M., Shima, T.: Virtual target approach for multi-evader intercept. In: 2022 European Control Conference (ECC), pp. 1491–1496 (2022). https://doi.org/10.23919/ECC55457.2022.9838427

[14] Giusti, A., Maffettone, G.C., Fiore, D., Coraggio, M., di Bernardo, M.: Distributed control for geometric pattern formation of large-scale multirobot systems. arXiv (2022). https://doi.org/10.48550/ARXIV.2207.14567. https://arxiv.org/abs/2207.14567

[15] Wang, S., Li, Z., Gao, H., Shan, K., Li, J., Yu, H.: Agile running control for bipedal robot based on 3d-slip model regulation in task-space. In: Liu, H., Yin, Z., Liu, L., Jiang, L., Gu, G., Wu, X., Ren, W. (eds.) Intelligent Robotics and Applications, pp. 505–516. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-13844-7_48

[16] Fong, P., Fang, C., He, J.: Optimal attack against coverage path planning in multi-robot system. In: Ren, Z., Wang, M., Hua, Y. (eds.) Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control, pp. 1760–1772. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-3998-3_164

[17] Parker, L.E.: Encyclopedia of Complexity and Systems Science. Multiple Mobile Robot Teams, Path Planning and Motion Coordination in, pp. 5783–5800. Springer, New York, NY (2009). https://doi.org/10.1007/978-0-387-30440-3_344. https://doi.org/10.1007/978-0-387-30440-3_344

[18] Kim, S.H., Yoon, S.G., Chae, S.H., Park, S.: Economic and environmental optimization of a multi-site utility network for an industrial complex. Journal of Environmental Management (2010). https://doi.org/10.1016/j.jenvman.2009.09.033

[19] Yahya, A., Li, A., Kalakrishnan, M., Chebotar, Y., Levine, S.: Collective robot reinforcement learning with distributed asynchronous guided policy search. In: IEEE International Conference on Intelligent Robots and Systems, pp. 79–86 (2017). https://doi.org/10.1109/IROS.2017.8202141

[20] Di Carlo, J., Wensing, P.M., Katz, B., Bledt, G., Kim, S.: Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In: IEEE International Conference on Intelligent Robots and Systems, pp.

1–9 (2018). https://doi.org/10.1109/IROS.2018.8594448

[21] Bjelonic, M., Grandia, R., Geilinger, M., Harley, O., Medeiros, V.S., Pajovic, V., Jelavic, E., Coros, S., Hutter, M.: Offline motion libraries and online MPC for advanced mobility skills:. https://doi.org/10.1177/02783649221102473 (2022). https://doi.org/10.1177/02783649221102473

[22] Lee, K.-H., Nachum, O., Zhang, T., Guadarrama, S., Tan, J., Yu, W.: PI-ARS: Accelerating Evolution-Learned Visual-Locomotion with Predictive Information Representations. arXiv (2022). https://doi.org/10.48550/ARXIV.2207.13224. https://arxiv.org/abs/2207.13224

[23] Das, A., Naroditsky, O., Zhu, Z., Samarasekera, S., Kumar, R.: Robust visual path following for heterogeneous mobile platforms. In: 2010 IEEE International Conference on Robotics and Automation, pp. 2431–2437 (2010). https://doi.org/10.1109/ROBOT.2010.5509699

[24] Debord, M., Hönig, W., Ayanian, N.: Trajectory planning for heterogeneous robot teams. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7924–7931 (2018). https://doi.org/10.1109/IROS.2018.8593876

[25] Saribatur, Z.G., Patoglu, V., Erdem, E.: Finding optimal feasible global plans for multiple teams of heterogeneous robots using hybrid reasoning: an application to cognitive factories. Autonomous Robots **43**(1), 213–238 (2019). https://doi.org/10.1007/s10514-018-9721-x

[26] Collins, J., Chand, S., Vanderkop, A., Howard, D.: A review of physics simulators for robotic applications. IEEE Access **9**, 51416–51431 (2021). https://doi.org/10.1109/ACCESS.2021.3068769

[27] Yu, Y., Miao, Z., Wang, X., Shen, L.: Cooperative circumnavigation control of multiple unicycle-type robots with non-identical input constraints. IET Control Theory and Applications **16**(9), 889–901 (2022). https://doi.org/10.1049/CTH2.12275

[28] Calderón-Arce, C., Brenes-Torres, J.C., Solis-Ortega, R.: Swarm Robotics: Simulators, Platforms and Applications Review. Computation **10**(6) (2022). https://doi.org/10.3390/computation10060080

[29] Garzón, M., Valente, J., Roldán, J.J., Garzón-Ramos, D., de León, J., Barrientos, A., del Cerro, J.: Robot Operating System (ROS). Studies in Computational Intelligence. Using ROS in Multi-robot Systems: Experiences and Lessons Learned from Real-World Field Tests, pp. 449–483. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54927-9_14

[30] Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L.M., Dorigo, M.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence **6**(4), 271–295 (2012)

[31] Fraisse, P., Perruquetti, W., Gil, A., Divoux, T., Zapata, R.: Stratégie de Commande Collaborative pour des Réseaux de Robots (2005). https://doi.org/10.1109/ICARCV.2004.1468982. http://jnrr05.irisa.fr/document/jnrr-002-01-fraisse.pdf

[32] Khamis, A., Hussein, A., Elmogy, A.: Cooperative Robots and Sensor Networks. Multi-robot Task Allocation: A Review of the State-of-the-Art, pp. 31–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18299-5_2

[33] Tkach, I., Edan, Y.: Automation, Collaboration, & E-Services. Distributed Heterogeneous Multi Sensor Task Allocation Systems. Automation, Collaboration, & E-Services, vol. 7, pp. 133–139. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-34735-2. http://link.springer.com/10.1007/978-3-030-34735-2

[34] Ball, M.G., Qela, B., Wesolkowski, S.: Recent Advances in Computational Intelligence in Defense and Security. Studies in Computational Intelligence. A Review of the Use of Computational Intelligence in the Design of Military Surveillance Networks, pp. 663–693. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-26450-9_24

[35] Otte, M., Kuhlman, M.J., Sofge, D.: Auctions for multi-robot task allocation in communication limited environments. Autonomous Robots **44**(3), 547–584 (2020). https://doi.org/10.1007/s10514-019-09828-5

[36] Tihanyi, D., Lu, Y., Karaca, O., Kamgarpour, M.: Multi-robot task allocation for safe planning against stochastic hazard dynamics. arXiv (2021). https://doi.org/10.48550/ARXIV.2103.01840. https://arxiv.org/abs/2103.01840

[37] Joseph, L.: Mastering ROS for Robotics Programming : Design, Build, and Simulate Complex Robots Using Robot Operating System and Master Its Out-of-the-box Functionalities. Packt Publishing, ??? (2015). https://books.google.dz/books?id=AVX5jgEACAAJ