2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Online Coverage Path Planning Scheme for a Size-Variable Robot

M. A. Viraj J. Muthugala, S. M. Bhagya P. Samarakoon, and Mohan Rajesh Elara

Abstract— **Coverage Path Planning (CPP) is an essential feature of robots deployed for applications such as lawn mowing, cleaning, painting, and exploration. However, most of the state-of-the-art CPP methods are proposed for fixed-morphology robots, and the coverage performance is limited by physical constraints such as the inaccessibility of narrow spaces. Apart from area coverage, productivity depends on coverage time and energy usage. A robot capable of varying its footprint size could be a solution for improving productivity in these aspects. In addition to that, the environments, where robots are deployed for coverage, are often subjected to changes causing uncertainties. Therefore, this paper proposes an online CPP scheme for a size-variable robot to improve coverage productivity. The navigation planning of the proposed Size-Variable CPP (VSCPP) scheme has been implemented by adapting a Glasius bio-inspired neural network that guides a robot in an efficient path for coverage while coping with dynamic changes. The size variation required for a situation is determined by analyzing a set of occupancy grid maps corresponding to the size steps of the robot. According to the results, the proposed VSCPP can ascertain coverage while coping with dynamic changes in an environment. The reduction of the coverage time due to the size variability is significant compared to a robot with no VSCPP scheme.**

I. Introduction

In Coverage Path Planning (CPP) problems, a robot is expected to navigate and cover all areas designated to access in a given environment [1]. CPP is a crucial feature for robots developed for coverage demanding applications, such as painting [2], cleaning [3], lawn mowing [4], exploration [5], and agriculture [6]. The productivity of these robotic applications eventually depends on the area coverage, coverage time, and energy efficiency [7]–[9].

Technological advancements in sensor perception, navigation, and computational intelligence have paved the way for developing many CPP algorithms for robotics applications [1], [10]. The work [11] discussed CPP by addressing the problem of chaotic path planning. A cell permeability-based approach to reduce overlapping in CPP has been introduced in [12]. A memory technique is applied in this work to avoid overlapping cells during the coverage. Robots deployed for most coverage applications have to operate in obstacle-cluttered environments. In this regard, the work [13] proposed a cellular decomposition approach that divides the environment into subregions for improving the coverage of a mobile robot deployed in obstacle terrains.

The obstacle layout of an environment is often subjected to alterations such as moving furniture. Therefore, robots intended for coverage applications should be able to cope with partially known and dynamic environments. In this regard, the work [14] introduced an adaptive CPP method inspired by the predator-prey relation for a mobile robot to cope with dynamic changes in an environment. Here, a reward function based on predator-prey relation is used to guide the robot at each step heuristically. A combination of the rolling window approach and the distance transform algorithm has been proposed for the CPP of cleaning robots in dynamic environments in [15]. This combined approach creates a local map from the global map for updating dynamic obstacles for CPP. According to the comparison in [16], neural network-based CPP methods are well suited for robots deployed in partially known or dynamic environments. For example, a Bio-inspired Neural Network (BNN) has the capacity to CPP in uncertain environments [17]. Computational and implementation complexities are concerns of online CPP algorithms, and the work [18] proposed a Discrete Bio-inspired Neural Network (DBNN) that has lower computational complexity and simpler implementation compared to a BNN for CPP. A Glasius Bio-inspired Neural Network (GBNN) is also a discrete-time Hopfield-type neural network that can be used for energy-efficient online CPP [19], [20].

Environments where robots are deployed for coverage, are often cluttered with unstructured obstacles. These obstacles can create inaccessible areas for a fixed-morphology robot due to physical constraints degrading the coverage performance [21]. Therefore, the performance of CPP methods discussed above is bounded by physical constraints since the methods have been designed for robots with fixed morphologies. This limitation could be overcome if a robot could vary its physical morphology based on the situation. In this regard, a class of hinged-reconfigurable robots that can vary the footprint shape have been introduced for floor cleaning applications [22], [23]. Furthermore, a concept regarding a hinged-reconfigurable robot that can change its size apart from the shape has been introduced in [24]. Here, the size is varied for subregion wise where the size is fixed for a subregion. These hinged-reconfigurable robots consist of a set of serially connected blocks through hinges that facilitate the relative motion of the blocks for shape-changing. This construction increases the cost, flimsiness, and complexity of fabrication, control, and autonomy of hinged reconfigurable robots (e.g., The locomotion system of the hinged robot used in [22] consists of 16 drive motors connected to omni

This research is supported by A*STAR under its Industry Alignment Fund – Pre Positioning (Award M21K1a0104), National Robotics Programme under its NRP BAU (Funding Agency Project No. M22NBK0054), and National Robotics Programme under its Robot Domain Specific (Funding Agency Project No. W1922d0110) and administered by the Agency for Science, Technology and Research.

The authors are with the Engineering Product Development Pillar, Singapore University of Technology and Design, 8 Somapah Rd, Singapore 487372.

wheels.). Modular robots also have the strength to make transformations such as size and shape changing [25]. The transformations are performed by assembling and dissembling the modules [26], [27]. Nevertheless, the deployment cost of modular multi-robots would also be high since each robot module should have locomotion modules, processing, perception, communication abilities, and task-related payload.

A design of a robot that can vary its footprint size has been introduced in [28]. The work showed that the ability to vary the size of the robot could improve the coverage performance in terms of area coverage and coverage time, presenting preliminary results. Since this robot is a single module that can expand and shrink, the cost of deployment is lesser, and reliability is higher compared to a modular robotic system with multiple modules or a hinged reconfigurable robot. Nevertheless, the main focus of the work is limited to a proposal of the hardware design aspects and the control of a size variable robot. A zigzag-motion-based CPP method has been used here to validate the possible usability of the proposed robot design. The used zigzag-motion-based CPP method is inefficient and incapable of coping with dynamic changes in an environment, which is crucial for a robot deployed for coverage applications.

Therefore, this paper proposes a novel online CPP scheme for a size-variable robot. The proposed Variable-Size CPP (VSCPP) is designed by adapting a GBNN that efficiently guides the robot and effectively varies the size for coverage while coping with dynamic changes in the environment to improve productivity. Furthermore, this paper is the first to report the usage of a neural network based CPP scheme for a size-reconfigurable robot. An overview of the proposed system is given in Section II. The proposed VSCPP scheme is detailed in Section III. Particulars on validating the behavior and the performance of the proposed scheme are discussed in Section IV. The concluding remarks are given in Section V.

II. System Overview

The size of a robot could alter the performance aspects of a coverage application. If a robot was fixed to a large size, it could not access the narrow areas resulting in lower coverage. In contrast, if the robot was fixed to a small size, the robot could access the narrow areas and maximizes the coverage. Nevertheless, the small robot requires many steps to cover the large free areas, resulting in a higher coverage time. Both coverage amount and coverage time are crucial performance factors for coverage applications. Therefore, the main aim of the proposed VSCPP scheme is to utilize the size variation ability of a robot to improve coverage performance in terms of area coverage and coverage time. An overview of the proposed VSCPP scheme is depicted in Fig. 1. A robot that can vary its footprint size through reconfiguration is considered here, where the robot can change its size from the smallest to the largest in N steps. M_i is the width of the robot in the i^{th} size step where M_1 represents the lower bound of the width, while M_N is the upper bound.

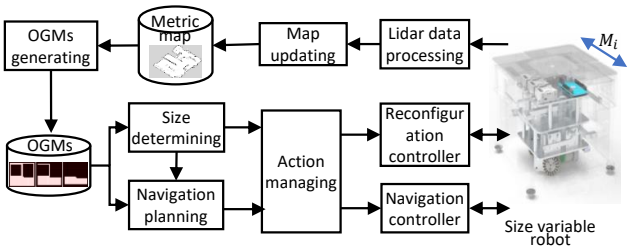

Fig. 1. System overview.

Through the Lidar data of the robot, the system can perceive the environment to detect any unknown change and subsequently update the metric map. The lidar data processing module and the map updating module are responsible for this regard. A set of Occupancy Grid Maps (OGMs) corresponding to the size steps of the robot are created by the OGMs generating module by taking the metric maps and the robot size variation capability. The navigation planning module directs the robot navigation to ascertain the complete area coverage efficiently. A Glasius Bio-inspired Neural Network (GBNN) [20] is adapted for realizing the navigation planning module. The set of OGMs generated per the robot size steps is used by the size determining module to decide the size variation of the robot. The size determining module is implemented in such a way that it allows the robot to cover open free areas with larger sizes while accessing narrow areas with smaller sizes by utilizing the size variation ability of the robot. The action managing module is responsible for coordinating the two main behaviors of the proposed VSCPP; navigation and size variation of the robot. Per the incoming instructions from the action managing module, the navigation controller performs the low-level functionalities required for the navigation, such as localization and drive-wheel activation. Similarly, the reconfiguration controller handles the activation of low-level functions for varying the size per the request of the action managing module.

III. Variable-Size Coverage Path Planning

A. Occupancy Grid Maps for Different Size Steps

The overall flow of the proposed Variable-Size Coverage Path Planning (VSCPP) scheme is given in Algorithm 1. A metric map of the environment created based on Lidar information is considered initially. The map can be fully known or partially known. Then, N number of obstacle inflated maps corresponding to the size steps are generated as shown in Fig. 2(a). The amount of inflation, L corresponding to the i^{th} size step of the robot, is taken as $M_i/2 - S/2$, where S is the cell size such that $S < M_1$.

The OGMs corresponding to these obstacle-inflated maps are then generated considering a grid with uniform cells. The OGMs corresponding to the obstacle-inflated grid maps are shown in Fig. 2(b). The OGM corresponding to the i^{th} size step is defined as OGM_i. The cells partially or fully occupied by the obstacles, including inflated regions, are considered obstacle cells. The other cells are considered free cells, where the robot is expected to visit them for coverage. The amount

Algorithm 1: Variable-Size Coverage Path Planning (VSCPP) Scheme

input : Metric map
output: Coverage path and robot size variation
Generate obstacle inflated maps;
Generate $OGM_i|i = 1 : N$;
Generate the neural map based on OGM_1;
p = initial position;
PSS = initial size step;
while *(There are cells to be visited)* **do**
  Tag p and neighbor cells per PSS as covered;
  Update neural activities;
  Select n based on (4);
  NSS = Size_Determine(n);
  if $NSS < PSS$ **then**
    Change size to NSS;
    Navigate to n;
  else if $NSS > PSS$ **then**
    Navigate to n;
    Change size to NSS;
  else
    Navigate to n;
  end
  $p = n$;
  $PSS = NSS$;
  if *Lidar information perceives a map change* **then**
    Update the metric map and $OGM_i|i = 1 : N$;
    Update neural map;
  end
end

Function `Size_Determine(`n`)`**:**
  for $(i = 1 : N)$ **do**
    if n^{th} *cell in* OGM_i *is free* **then**
      $NSS = i$;
    end
  end
  Return NSS;
end function

of inflation avoids the possible collision of the robot even when the robot center is on a free cell closest to an obstacle in the corresponding OGM with the specified size. Thus, the robot configured to the size M_i is on any free cell on OGM_i, the robot does not collide with any obstacle.

As seen in the OGMs depicted in Fig. 2(b), when the robot size is increased, the area of free cells where the robot can visit for coverage reduces. Moreover, the area coverage is reduced if the robot size increases. On the other hand, as the size of the robot gets large, the robot footprint covers many cells in the occupancy grid maps compared to the case of a smaller size where the robot could cover the area in fewer steps and time. Therefore, the proposed VSCPP scheme use size changeability to maximize the area coverage while reducing the coverage time. In this regard, the OGMs corresponding to the size steps of the robot is used to determine the maximum possible size to cover a

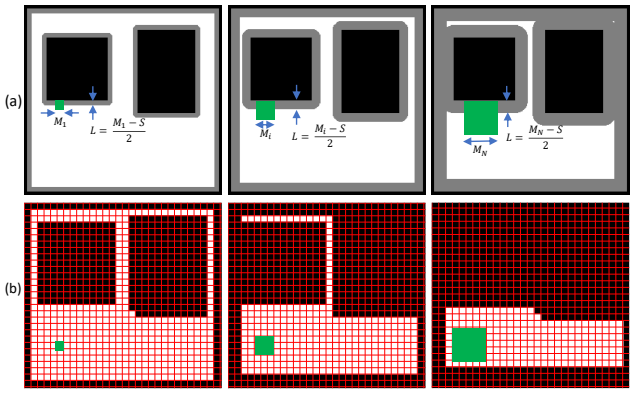

Fig. 2. Creating occupancy grid maps corresponding to different sizes of the robot. (a): Amounts of obstacle inflation per the size configuration of the robot. (b): The corresponding OGMs. The areas in black represent obstacles. Inflated areas are represented in gray. The robot is depicted in green.

particular set of cells. The maximum possible coverage could be achieved by using the smallest size of the robot and the corresponding OGM (i.e., OGM_1). Therefore, OGM_1 is used for the navigation behavior of the robot.

B. Navigation and Size Variation

The navigation planner of the coverage scheme uses a GBNN since it can be used to perform an efficient coverage while coping with dynamic changes in the environment [19], [20]. A GBNN is a topographically ordered Hopfield-type neural map that can represent the 2-dimensional working environment of a robot. A GBNN corresponds to OGM_1 of a working environment of the robot, where the p^{th} neuron in the GBNN represents the p^{th} cell of OGM_1, is created initially. This representation is depicted in Fig. 3, considering an example situation. The created GBNN is used to determine the navigation behavior required for the robot to cover the whole workspace.

The neurons in the GBNN have lateral connections that pass the neural activities. The strength of neural activity passing depends on the corresponding weight of the connection. It is considered that the weight of the connections depends on the Euclidean distance between the corresponding two neurons, where the strength of neural activity passing is exponentially decaying with the distance. Furthermore, a neuron receives the neural activities only from the neighborhood neurons that are within the receptive field of the

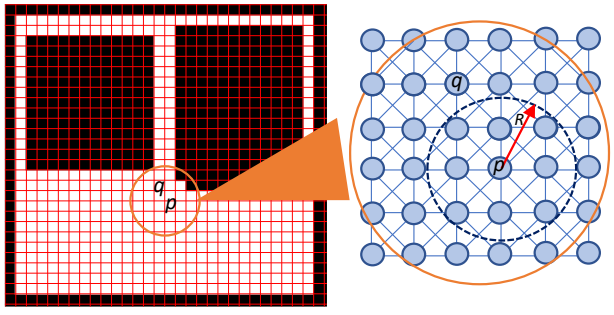

Fig. 3. Analogy between the environment and the neural map. (a) OGM_1 of the environment. (b) The GBNN corresponding to the grid map.

neuron. The connection weight, $w_{p,q}$ between the p^{th} and the q^{th} neurons can be formulated as in (1) where $|p-q|$ is the Euclidean distance between the p^{th} and the q^{th} neurons, R is the size of the receptive field, and alpha is a positive scalar constant.

$$w_{p,q} = \begin{cases} e^{-\alpha|p-q|^2} & \text{if } 0 < |p-q| \leq R \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In addition to the lateral connections, an external input that depends on the state of the corresponding cell is connected with each neuron. This input can have three different states where excitatory, inhibitory, and neutral inputs are offered for uncovered, obstacle, and covered cells, respectively. The assignment of the external input of the p^{th} neuron, E_p is given in (2), where V is a large positive scalar constant.

$$E_p = \begin{cases} +V & \text{if p}^{\text{th}} \text{ cell is uncovered} \\ -V & \text{if p}^{\text{th}} \text{ cell is obstacle} \\ 0 & \text{if p}^{\text{th}} \text{ cell is covered.} \end{cases} \tag{2}$$

Only the positive neural activities are propagated through the lateral connections to attract the robot toward the uncovered cells and repulse from obstacle cells. The neural activity of the p^{th} neuron at the $(t+1)^{\text{th}}$ time step can be defined as in (3) to accommodate this neural propagation behavior. Here, L is the number of neurons, and ψ is the activation function.

$$x_p(t+1) = \psi\left(\sum_{q=1}^{L} w_{p,q} \max\{x_q(t), 0\} + E_p\right) \tag{3}$$

where,

$$\psi(z) = \begin{cases} -1 & \text{if } z < 0 \\ \beta z & \text{if } 0 \leq z < 1 \\ 1 & \text{if } z \geq 1 \end{cases}$$

In each time step, the robot is moved to a new cell where the goal is to visit all the uncovered cells. The next cell to visit, n is selected as in (4) among the candidate cells considering the neural activity and the energy usage. Here, x_c represents the neural activity of the c^{th} candidate cell within the receptive field of the present cell (defined as p), and $\tilde{e_{c,p}}$ is the normalized energy for distance travel and direction change required to drive the robot to the c^{th} cell from the present cell. The attraction toward uncovered cells and energy requirement is counteracted by η, where η is adapted per the context to avoid the deadlocks. The scalar constant, $H \in [0,1]$ can be defined per performance expectations. The consideration of neural activity directs the robot toward visiting the uncovered cells, while the energy consideration paves the way to finding an efficient path.

$$n = \underset{\forall c \text{ within } R}{\arg\max} \left(x_c + \eta(1 - \tilde{e}_{c,p})\right) \tag{4}$$

where,

$$\eta = \begin{cases} 0 & \text{if } 1 \notin \{x_c\} \\ H & \text{otherwise} \end{cases}$$

After deciding on the next cell to visit (i.e., n), the VSCPP scheme should determine the suitable size-adjustment of the robot. In this regard, the algorithm first identifies the collision-free maximum size step of the robot (defined as NSS) to be on the n^{th} grid cell. This size step is identified by checking the status of the n^{th} cell in OGM_i for $i = 1$ to N. The size step corresponding to the maximum size OGM, where the n^{th} grid cell is free, is selected as NSS. If NSS is lesser than the present size step (defined as PSS), the robot has to first reduce its size by reconfiguring and then move to the n^{th} cell. This pre-size reduction allows the robot to move to the new location narrower than the present location without collisions. If the size step identified for the n^{th} cell is higher than the present size step (i.e., $NSS > PSS$), the robot first has to move the n^{th} cell and then reconfigure to the new size where the robot should expand. The post-size expansion would avoid the possible collision if the size expanded when the robot was in the present cell. When there is no size step difference in the present and following locations (i.e., $NSS = PSS$), the robot should move to the new location without any size variations.

After completing the moving to the n^{th} cell and size varying (if required), the n^{th} cell becomes the present cell, p and PSS becomes NSS. The effective cell coverage varies with the size of the robot. For example, the neighboring cells could be covered when the robot size step is high or vice versa (see Fig. 2(b)). Thus, the new present cell p is tagged as covered along with the neighboring cells, which corresponds to PSS. When a cell is tagged as covered, the external input for that neuron becomes zero. This process is continued until the neural activities of all the neurons in the GBNN becomes zero (i.e., covering all the free cells in OGM_1).

The decision-making process of the GBNN-based navigation planner and the size-determining process are online. Upon detecting a change in the environments through the Lidar information, the OGMs and the neural map are updated. Subsequently, this update facilitates the adaptation of the proposed VSCPP to a new situation.

IV. Results and Discussion

A. Experimental setup

The size-variable robot introduced in [28] has been considered to validate the behavior and performance of the proposed VSCPP method through Matlab based simulations. The considered robot can vary its width between 20 cm to 30 cm (i.e., $M_1 = 20$ cm, $M_N = 30$ cm) using its reconfiguration ability, and it can vary the size in the full range within 7 s. Furthermore, the robot consists of an omni drive mechanism for the locomotion. In the validation, it was assumed that the robot could vary the size in two steps (i.e., $N = 2$) to par with the size range of the robot. The cell size of the OGMs (i.e., S) was configured to 8 cm considering the robot size, computational efficiency, and performance. The parameters of the navigation planner were configured as follows; $\alpha = 2$, $\beta = 0.7$, $R = 2$, $V = 100$, and $H = 0.5$.

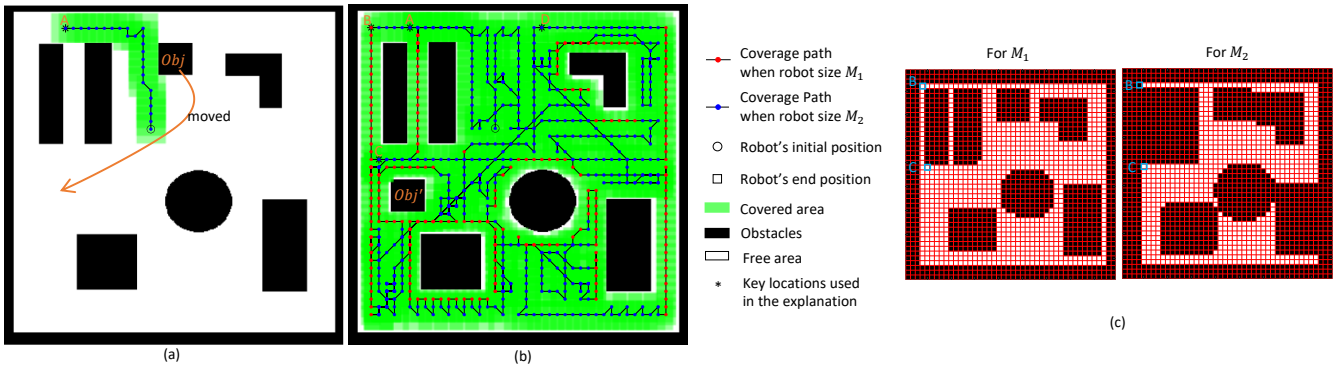

Fig. 4. Robot coverage behavior. (a): The robot coverage until the obstacle was moved to the new location, (b): The complete robot path and the coverage in the final arrangement of the environment, and (c): OGMs at the beginning of the coverage.

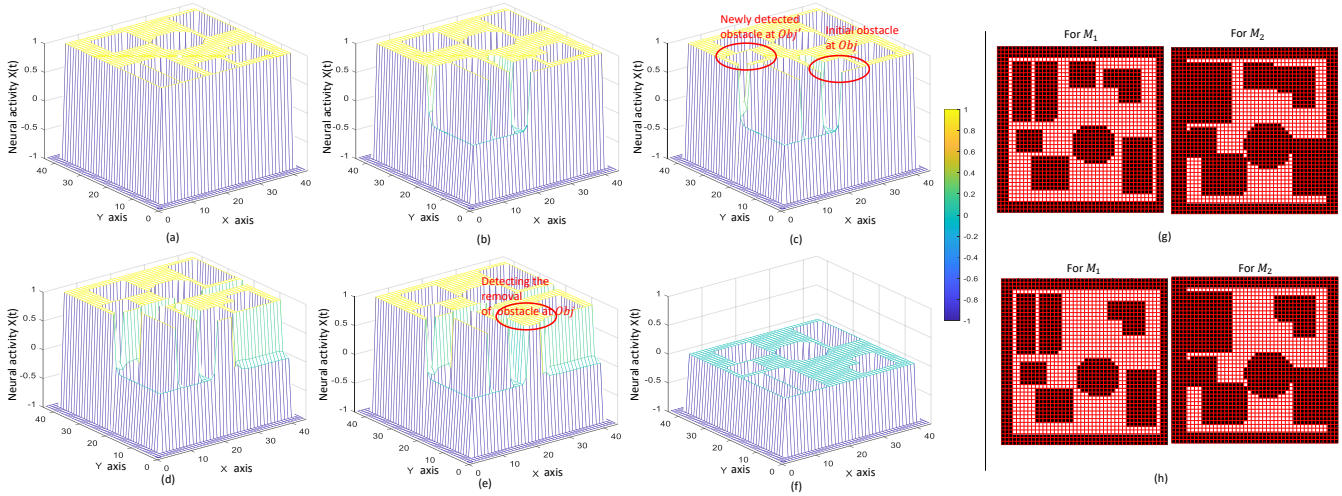

Fig. 5. Neural Activity (NA) and OGMs variation; (a): NA at the beginning, (b): NA before detecting the obstacle at Obj' (when robot reaching'C'), (c) NA after detecting the obstacle at Obj' (After reached 'C'), (d): NA before detecting the removal of the obstacle at Obj (when reaching 'D'), (e): NA after detecting the removal of the obstacle at obj (after reached 'D'), and (f): NA when finished the coverage. (g): OGMs after detecting the obstacle at Obj' and (h): OGMS after detecting the removal of the obstacle at obj.

B. Behavior of the VSCPP Scheme

An environment size of 3.4 m $\times$ 3.4 m with a dynamic change was considered for analyzing the behavior of the proposed VSCPP scheme. The robot coverage was initiated in the environment shown in Fig. 4(a). The OGMs corresponding to the environment at this instance is shown in Fig. 4(c). The neural activities of the neurons corresponding to the free cells of OGM_1 were high (i.e., 1.0), while obstacles had negative neural activities (see Fig. 5(a)). The robot started to perform the navigation using the size M_2 (largest size) since the corresponding cells of OGM_2 were also free. When the robot reached location 'A', the obstacle at Obj was being moved to location Obj'. The environment after the change can be seen in Fig. 4(b).

The robot continued the coverage without knowing the change. When the robot decided to move to location 'B', the size of the robot was varied to M_1 and continued the coverage from 'B' to 'C', which was a narrow path, maintaining the small size (i.e., M_1) since the corresponding cells of OGM_2 were not free (see Fig. 4(c)). When the robot reached 'C', it expanded the size to M_2 since the corresponding cell in OGM_2 was free. In addition, the robot discovered the change in the environment as adding a new obstacle to location Obj'. However, the robot was unaware that it was the same obstacle in location Obj. Hence a new obstacle was added to the maps while keeping the obstacle at the original location. The corresponding OGMs are shown in Fig. 5(g). The change in the neural activities due to the detection of this change can be seen comparing Fig. 5 (b) and (c) (A new region with negative neural activities was created).

The robot continued the coverage by varying its size per the information of the current OGMs until it reached 'D'. After reaching 'D', the robot discovered that the obstacle at Obj was removed and updated the OGMs (see Fig. 5(h)). Subsequent neural activity change corresponding to this discovery can be seen comparing Fig. 5 (d) and (e), where the negative activity region corresponding to the original location of the obstacle moved was updated with excitatory activities. The robot continued the coverage, considering the neural activities while varying its size per the new OGMs. The neural activity of all the cells decayed to zero, as shown in Fig. 5(f), when the robot covered all the cells. In this case, the robot achieved 92% area coverage while the coverage time was 34 minutes. The observations of this case validate the utilization of the size-varying ability of the proposed VSCPP to effectively access narrow and wide-free areas for coverage while coping with changes in an environment.

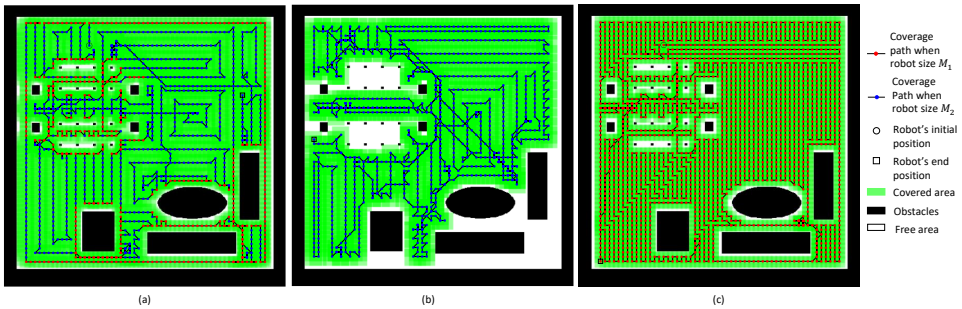

Fig. 6. Case 1 of the coverage comparison. (a): Coverage using the proposed VSCPP scheme, (b): Coverage when the robot size is fixed to M_2, and (c): Coverage when the robot size is fixed to M_1

C. Performance Evaluation

Ten heterogeneous test environments, such as living rooms, lobbies, and bedrooms, have been considered for evaluating the performance of the proposed VSCPP. Dynamic changes were induced to five of these test environments while keeping the other five static. In each environment, three cases were considered: the robot with the proposed VSCPP scheme, the robot fixed to the smallest size, and the robot fixed to the largest size. The area coverage and coverage time of each run were taken as the performance evaluation factors.

A layout of a living room was considered the first test environment. The coverage results obtained for the first test environment are shown in Fig. 6 (as sample results). In the first case, the robot with the proposed VSCPP scheme was considered. This case resulted in an area coverage of 94% (see Fig. 6(a) for coverage map), while the coverage time was 59 minutes. In contrast, the robot fixed to the largest size could achieve only 78% (see the coverage map in Fig. 6(b)), which is considerably lower than the proposed VSCPP scheme. The main reason for this lower coverage performance was the inability to access narrow spaces by the robot due to its large fixed size. On the other hand, the robot fixed to the smallest size achieved 92% area coverage (see Fig. 6(c)), comparable to the coverage that resulted from the robot with the proposed VSCPP scheme. Since the robot size was small, the robot could easily access narrow areas and cover much of the environment. However, the coverage time of this case was 87 minutes which was considerably higher than that of the robot with the VSCPP. The robot had to navigate many times for coverage since it can cover a limited area due to its small size resulting the poor performance in terms of coverage time (see Fig. 6(c)). The proposed VSCPP can resolve these two limitations since it can cover large free areas using a large size while accessing the narrow areas in a smaller size. Thus, improved performance has been observed from the proposed VSCPP in this environment in both coverage and coverage time.

Similar performance in area coverage and coverage time could be observed in all the test environments. The robot with the proposed VSCPP scheme could achieve a mean area coverage of 93.4% ($\pm$3.4) with a mean coverage time of 42.5 minutes ($\pm$22.3). The mean coverage observed from the robot fixed to the largest size is 69.4% ($\pm$11.9), while the robot fixed to the smallest size could achieve a mean coverage of 92.9% ($\pm$3.6) with a mean coverage time of 62.2 minutes ($\pm$30.7). A one-way ANOVA test and a post-hoc Tukey test conducted on the coverage time concluded that the robot fixed to the largest size had a significantly reduced area coverage ($F_{2,27} = 34$, $p < 0.05$). No significant difference between the area coverages of the robot with the VSCPP and the robot fixed to the smallest size was observed. Nevertheless, the robot fixed to the smallest size required significantly higher coverage time compared to the robot with the VSCPP (Per t-test, $t_9 = -6.10$, $p < 0.05$). Moreover, the proposed VSCPP scheme can reduce the mean coverage time by 32% while maintaining the same coverage performance. These results confirm that the proposed VSCPP scheme is capable of improving the productivity of coverage applications in terms of area coverage and coverage time while coping with changes in the environment.

V. Conclusions

This paper proposed an online VSCPP scheme for a size-variable robot that allows the robot to access narrow areas with small sizes while free areas in large sizes. The proposed VSCP scheme has two core modules: navigation planning and size determining. The navigation planning module has been developed by adapting a GBNN which can guide a robot in an efficient path for coverage while coping with changes in the environment. A set of OGMs generated corresponding to the size steps of the robot is used for determining the required size for accessing cells.

The proposed VSCPP scheme has been validated considering an existing design of a size-variable robot. According to the observed behavior of the proposed VSCPP scheme, the proposed VSCPP is effective in appropriately varying the size of the robot to access narrow and wide free areas for coverage while coping with the dynamic changes in environments. In terms of performance, the proposed VSCPP scheme can significantly reduce the coverage time while maximizing the area coverage compared to a robot without a VSCPP scheme. Therefore, the proposed VSCPP would be of great interest to the development of robots targeted for coverage applications. For future work, the VSCPP scheme will be enhanced to determine optimum size variation and navigation to maximize performance in terms of area coverage, coverage time, and energy.

REFERENCES

[1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[2] E. Asadi, B. Li, and I.-M. Chen, "Pictobot: A cooperative painting robot for interior finishing of industrial developments," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 82–94, 2018.

[3] H. Liu, J. Ma, and W. Huang, "Sensor-based complete coverage path planning in dynamic environment for cleaning robot," *CAAI Transactions on Intelligence Technology*, vol. 3, no. 1, pp. 65–72, 2018.

[4] I. Daniyan, V. Balogun, A. Adeodu, B. Oladapo, J. K. Peter, and K. Mpofu, "Development and performance evaluation of a robot for lawn mowing," *Procedia Manufacturing*, vol. 49, pp. 42–48, 2020.

[5] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Gaussian processes autonomous mapping and exploration for range-sensing mobile robots," *Autonomous Robots*, vol. 42, no. 2, pp. 273–290, 2018.

[6] I. A. Hameed, "Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3, pp. 965–983, 2014.

[7] K. Zheng, G. Chen, G. Cui, Y. Chen, F. Wu, and X. Chen, "Performance metrics for coverage of cleaning robots with mocap system," in *International conference on intelligent robotics and applications*. Springer, 2017, pp. 267–274.

[8] S. Rhim, J.-C. Ryu, K.-H. Park, and S.-G. Lee, "Performance evaluation criteria for autonomous cleaning robots," in *2007 International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 2007, pp. 167–172.

[9] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, and M. R. Elara, "Tradeoff between area coverage and energy usage of a self-reconfigurable floor cleaning robot based on user preference," *IEEE Access*, vol. 8, pp. 76 267–76 275, 2020.

[10] Y. Kang and D. Shi, "A research on area coverage algorithm for robotics," in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*. IEEE, 2018, pp. 6–13.

[11] E. Petavratzis, L. Moysis, C. Volos, M. K. Gupta, I. Stouboulos, and S. Goudos, "Chaotic motion control of a mobile robot using a memory technique," in *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2020, pp. 506–511.

[12] K. Guruprasad and T. Ranjitha, "Cpc algorithm: Exact area coverage by a mobile robot using approximate cellular decomposition," *Robotica*, vol. 39, no. 7, pp. 1141–1162, 2021.

[13] C. Li, Z. Wang, C. Fang, Z. Liang, Y. Song, and Y. Li, "An integrated algorithm of ccpp task for autonomous mobile robot under special missions," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 1357–1368, 2018.

[14] M. Hassan and D. Liu, "Ppcpp: A predator–prey-based approach to adaptive coverage path planning," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 284–301, 2019.

[15] Y. Zhou, R. Sun, S. Yu, Y. Sun, and L. Sun, "A complete coverage path planning algorithm for cleaning robots based on the distance transform algorithm and the rolling window approach in dynamic environments," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2017, pp. 1335–1340.

[16] R. Bormann, F. Jordan, J. Hampp, and M. Hägele, "Indoor coverage path planning: Survey, implementation, analysis," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1718–1725.

[17] K. Chen and Y. Liu, "Optimal complete coverage planning of wall-climbing robot using improved biologically inspired neural network," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2017, pp. 587–592.

[18] J. Zhang, H. Lv, D. He, L. Huang, Y. Dai, and Z. Zhang, "Discrete bioinspired neural network for complete coverage path planning," *International Journal of Robotics and Automation*, vol. 32, no. 2, 2017.

[19] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, and M. R. Elara, "Toward energy-efficient online complete coverage path planning of a ship hull maintenance robot based on glasius bio-inspired neural network," *Expert Systems with Applications*, vol. 187, p. 115940, 2022.

[20] B. Sun, D. Zhu, C. Tian, and C. Luo, "Complete coverage autonomous underwater vehicles path planning based on glasius bio-inspired neural network algorithm for discrete and centralized programming," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 1, pp. 73–84, 2018.

[21] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, and M. R. Elara, "Design by robot: A human-robot collaborative framework for improving productivity of a floor cleaning robot," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7444–7450.

[22] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, R. E. Abdulkader, S. W. Si, T. T. Tun, and M. R. Elara, "Modelling and control of a reconfigurable robot for achieving reconfiguration and locomotion with different shapes," *Sensors*, vol. 21, no. 16, p. 5362, 2021.

[23] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, and M. R. Elara, "Online complete coverage path planning of a reconfigurable robot using glasius bio-inspired neural network and genetic algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5744–5751.

[24] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, M. R. Elara, *et al.*, "Toward pleomorphic reconfigurable robots for optimum coverage," *Complexity*, vol. 2021, 2021.

[25] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[26] J. Seo, J. Paik, and M. Yim, "Modular reconfigurable robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 63–88, 2019.

[27] D. Zhao and T. L. Lam, "Snailbot: a continuously dockable modular self-reconfigurable robot using rocker-bogie suspension," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4261–4267.

[28] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, M. Kalimuthu, S. K. Chandrasekaran, and M. R. Elara, "Design of a reconfigurable robot with size-adaptive path planner," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 157–164.