Proceedings of The National Conference
On Undergraduate Research (NCUR) 2019
Kennesaw State University
Kennesaw, Georgia
April 11-13, 2019

Marsupially-Aided Robotic Snake Exploration and Navigation of Cluttered Environments

Alexander Faché, Nelson Raphael, Alicia Mora Velasco, Elizabeth Prucka,
João Françolin, Gabrielle Duva, Alisa Allaire
Department of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Faculty Advisor: Dr. Patricio A. Vela
Graduate Supervisor: Alexander H. Chang

Abstract

Environments characterizing disaster zones frequently impose locomotive limitations on traditionally larger, wheeled or tracked mobile platforms due to the inability to drive over large obstacles and a lack of precise maneuverability. Alternative robotic solutions demonstrate potential to overcome many of these limitations; snake-like robots, in particular, are advantaged in narrow, confined spaces, desert-like terrain as well as arboreal environments, among others that may be encountered in disaster zones comprising fallen rubble and blocked paths. Larger, wheeled robots, on the other hand, possess high-speed mobility useful for expedient traversal of large, obstacle-free expanses of terrain. The latter additionally support greater carrying capacity, allowing for more capable on-board computing as well as power capacity. We integrate instances of each class of mobile robots into a marsupial robotic pair that, together, leverages their individual strengths while limiting the drawbacks of either, to accomplish autonomous exploration in a planar, obstacle-cluttered environment. The marsupial pair, and accompanying robotic architecture we develop, are challenged to explore and navigate an arbitrary obstacle configuration in a planar indoor environment. In a collaborative manner that utilizes each platform according to its strengths, the Turtlebot and robotic snake employ SLAM and frontier exploration algorithms, in conjunction with locomotive primitives available to each, to cooperatively explore, map, and navigate an initially unknown scenario.

Keywords: Marsupial Robotics, Snake-like Robot, Turtlebot, Exploration, Navigation, SLAM

1. Background

Use of mobile robots in exploration and search and rescue (SAR) applications entails a variety of challenging terrain that must be traversed. SAR robots serve as the eyes and ears of human operators, capturing sensory information during exploration that may, for instance, reveal victim locations in highly-cluttered disaster zones while ameliorating risk posed to rescue personnel[1]. At the most basic level, autonomous robots operating in these environments must possess the ability to competently navigate and traverse initially unknown, obstacle-dense scenarios in order to facilitate high-level mission objectives such as assessment of structural integrity as well as exploration and mapping of areas where victims may be stranded. An analysis of past deployments of SAR robots has revealed prevalent issues in their design for SAR applications. A feature characterizing locomotive scenarios in both natural terrain as well as disaster sites is the presence of initially unmodeled, arbitrary obstacle arrangements. Successful exploration entails circumvention of these impediments as they are discovered[2]. Often, however, no single vehicle may be suitably-equipped to operate in all varieties of terrain characterizing disaster scenarios. Wheeled and tracked vehicles are impeded in terrain littered with obstacles that are large in size relative to the robot itself. Their large footprint additionally precludes traversal of narrow passageways to completely explore a scenario.

Biologically-inspired locomotive mechanisms, such as legged or snake-like robots are more appropriately applied to operate in these challenging locomotive environments; however, they are a less effective solution for mobility over open, clear terrain.

We pair a snake-like robot with a Turtlebot, a wheeled differential drive platform, to operate cooperatively as a marsupial pair. The Turtlebot is equipped to tow the robotic snake across large, clear expanses of terrain; the robotic snake is adept at traversing narrow passageways, to which the Turtlebot is denied access. A marsupial robotic software architecture is developed for this system that guides the two mobile robots to autonomously explore and navigate an initially un-modeled, unknown environment cluttered with obstacles. In a final demonstration, the pair of robots successfully explore and map an indoor environment, leveraging their individual strengths based upon characteristics of the local region requiring exploration.

2. Literature Review

An ability to intelligently negotiate complex and initially un-modeled terrain is a fundamental prerequisite for mobile robots participating in search and rescue missions within disaster zones[3]. However, no single mobile robotic platform is adequately suited for the variety of terrain entailed in such scenarios; task completion may require competent locomotion through restricted corridors, inconveniently positioned access points, and even large open expanses, among other challenges. Traditional wheeled robots often struggle with variably-sized, rubble-like terrain[4]; aerial drones may be denied from low altitude access points and low-ceiling cavities. Treaded robots, while adept on loose, rubble-like terrain, are vertically challenged, as are most traditional, rover-like mechanisms[2]. Modular robots and robotic snakes are endowed with many degrees of freedom (DOFs) as well as small profiles, permitting admittance to small gaps and constricted passage ways[5]. Locomotion on these robotic mechanisms has focused on a handful of biologically-inspired gaits, each advantaged in different locomotive scenarios. These include lateral undulation, sidewinding, accordion-like concertina motion, and rectilinear motion[6]. Of these, rectilinear modes of locomotion are well-suited to traversal of tightly confined spaces due to the elongated, narrow footprint they occupy. Additionally, during rectilinear motion, the importance of snakes actually lifting parts of their bodies may lead to a reduction in energy loss driven by ground friction[7]. Travel over large expanses of clear terrain, however, is not a locomotive scenario that snake-like robots and their associated gaits are advantaged in relative to larger, wheeled platforms. Rectilinear gait modes, in particular, are even disadvantaged with respect to many other gaits often utilized by snake-like robotic platforms[8]. Additional locomotive limitations for this class of robots are driven by restrictions to onboard power and onboard computing, due to low payload capacity and small form factor[9].

Marsupial robotic systems draw inspiration from marsupial relationships found in nature; distinctly advantaged mobile platforms are paired such that the resulting collaborative relationship exploits complementary strengths of both platforms while shedding limitations and weaknesses belonging to each. Such systems present a viable solution to many problems faced by traditional, single robot systems applied to SAR tasks in complex environments. Two or more mobile robots leverage their collection of unique strengths to increase the scope of applicable operational environments the system as a whole may competently operate in. Marsupial robotic systems are distinguished from other multi-robot systems by the presence of a larger container robot that carries one or more smaller passenger robots. The strengths of the container lies in perception, computational power, and long-distance travel capacity. The passenger is specialized to the demands of specific environments and tasks[10]. A collective of individually distinct robots presents the opportunity for enhanced task performance, increased task reliability and decreased cost relative to traditional robotic systems[11].

3. Methodology

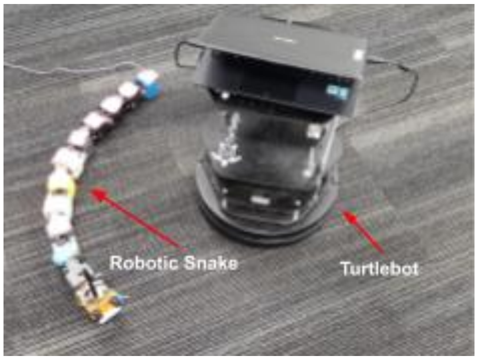

Figure 1. A robotic snake (passenger) and a Turtlebot (container) comprise a marsupial robotic pair for exploration of obstacle-strewn scenarios.

3.1 Marsupial Container

In the context of planar exploration in obstacle-dense scenarios, a Turtlebot is employed (Figure 1, right) as the marsupial container. The Turtlebot is a unicycle-like mobile robot; it is able to move forward, backward, and rotate in place[12]. It is equipped with onboard processing power (2013 MacBook Pro) and the carrying capacity to transport a robotic snake over large expanses of clear terrain. However, it is not suited to travel over rugged, complex terrain and dense obstacle arrangements.

The model used is a TurtleBot2 equipped with a Kinect camera. Depth information characterizing the surrounding environment is captured by the Kinect sensor; this informs a Simultaneous Localization And Mapping (SLAM) algorithm applied to incrementally generate a map of an initially unknown environment while simultaneously localizing the robot's location in that map. A frontier exploration algorithm, developed in Python, selects the points of interest that require exploration, in order to expand the robot's map of the world; an underlying Robot Operating System (ROS)[13] navigation framework is utilized to plan and track paths to points of interest. Frontier exploration continues until there are no accessible areas left for the Turtlebot to uncover. During exploration, points of interest not reachable by the Turtlebot, due to an inability to physically fit through an access point, are delegated to the robotic snake for later exploration (Figure 2). The collaborative robot software architecture implementing this approach was developed and integrated with ROS, a flexible middleware framework supporting robotic development and operation.

3.2 Passenger

The robotic snake used here (Figure 1, left) is comprised of eleven joints, allowing a multitude of three dimensional body shapes to be accomplished.[8] Each joint comprises a *Robotis Dynamixel RX-28* motor. Adjacent joint axes are arranged orthogonal to one another; this allows every other joint to accomplish pitch motions while the complementary set produce yaw. A *Robotis OpenCM 9.04* controller and *OpenCM EXP485* expansion board facilitate serial communication between a laptop and the motors. Locomotion is driven by cyclic changes in the robot's body shape, called gaits; these generate unique body-ground interactions that yield a net locomotive displacement. In contrast to the Turtlebot, the robotic snake is advantaged with a caterpillar-like rectilinear locomotive gait facilitating a narrow locomotive footprint; this enables it to traverse narrow corridors and tight spaces that the former cannot. We design additional motion primitives to further increase the robotic snake's locomotive versatility and exploratory capability in obstacle-dense scenarios. Gaits were designed using Matlab as a prototyping environment and then transcribed to Python for integration into the ROS framework; this facilitated more streamlined runtime execution of the robot architecture.

Once the Turtlebot completes exploration of all reachable points of interest, the robotic snake dismounts to continue exploring regions the Turtlebot was unable to navigate to. The robotic snake uses a variant of ORB SLAM[14] to map the environment while travelling to a goal position specified by the Turtlebot. Sensory information needed for SLAM is provided by a stereo camera positioned on the head of the robotic snake. A *Blackbird 2 3D FPV Stereo Came*ra serves as the robot's eyes; a *FX X50-2 5.8G 200mW 40CH Wireless AV Transmitter* communicates video data to a receiving laptop controlling the robotic snake. The stereo camera is equipped with dual lenses separated by a known positional offset. Much like humans, this offset in left and right images seen by each eye allow the stereo camera to accurately sense depth. When the goal is reached by the robotic snake, it returns to the Turtlebot (Figure 2).

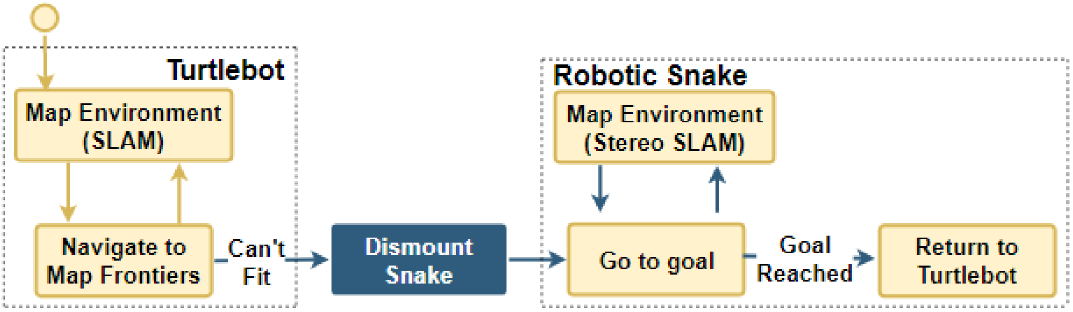

Figure 2. Block diagram of the marsupial relationship characterizing the Turtlebot and robotic snake during collaborative exploration.

3.3 Marsupial System

In pairing both robots, their strengths are leveraged when appropriate to the locomotive challenge at hand; their individual weaknesses are mitigated. In this manner the exploratory capabilities of the overall multi-agent system increases.

The marsupial pair is linked by a tether (Figure 1). This tether allows the Turtlebot to provide power and locomotive commands to the robotic snake.

Since the travel speed of the robotic snake is much slower than its marsupial container, a carriage prototype was designed for the robotic snake to be transported aboard the Turtlebot during the latter's exploration. The design facilitates a simplified mounting and dismounting process, allowing the robotic snake to curl around the Turtlebot for transport. Once the Turtlebot has explored all regions of interest, the robotic snake is dismounted from the carriage and leverages its narrow locomotive footprint to continue exploring regions inaccessible to the Turtlebot.

4. Marsupial Robotic Architecture

Both the Turtlebot and robotic snake platforms are integrated through a marsupial robotic exploration and navigation software architecture. This architecture expands the capabilities of each platform's individual potential. Both robots apply variants of SLAM to map their local environments during the course of exploring an initially unknown scenario. Subsequent navigation and robot travel proceed based on the incrementally uncovered map of the environment.

4.1 Turtlebot Exploration

The Turtlebot maps an unknown environment using the GMapping SLAM algorithm[15]. A Frontier Exploration algorithm is applied to the incrementally constructed map to detect frontiers and explore them. Frontiers are defined as boundaries between explored and unexplored regions of the map; these locations present opportunities to gain knowledge about the environment.

Using a laser scan abstraction of the Kinect depth sensor, the Turtlebot measures distance to nearby objects to update a map of its surroundings. Upon completing each scan and updating the map, frontiers in the environment are recomputed. Each frontier, $i \in \mathbb{Z}^{+}$, is characterized by its centroid (spatial) location, $\boldsymbol{x}_i{}^{frontier} \in \mathbb{R}^2$, and width, $w \in \mathbb{R}$. A geometrically-derived utility is additionally assigned based on each frontier's width and Euclidean distance from the Turtlebot, whose (spatial) position is denoted, $x^{TB} \in \mathbb{R}^2$. Utility is interpreted as a frontier's potential for knowledge gain and is computed as,

$$u_i = \frac{w}{\left\| x^{TB} - x_i^{frontier} \right\|_2}. \quad (1)$$

Frontiers are prioritized based on their utility, u_i; higher priority is associated with frontiers possessing greater utility. The greater the width of the frontier the greater the utility, as a large width correlates to a larger area to explore and greater potential for knowledge gain. Simultaneously, a shorter Turtlebot-to-frontier distance suggests the frontier is quickly reachable in order to acquire knowledge about the environment.

Turtlebot navigation goals are assigned to be the centroid location, $\boldsymbol{x}_i{}^{frontier}$, whose frontier has greatest utility, u_i. Upon goal assignment, the ROS navigation stack plans a feasible route through mapped regions of the environment, from the current Turtlebot location, x^{TB}, to the goal location. Once a goal has been reached, the associated frontier is no longer relevant. The list of frontiers is then recomputed based on an updated map and prioritized; the frontier with greatest utility is again selected as the next goal to navigate and travel to. This process continues until all frontiers, reachable by the Turtlebot, have been explored and no others remain.

During frontier exploration some goals that were selected for Turtlebot exploration may have been unreachable; this may occur if the only feasible paths to the goal require traversal of passageways too narrow for the footprint of the Turtlebot. This path planning failure is detected by the ROS navigation framework when attempting to plan a route to a goal. In these situations, the associated frontier is removed from consideration and instead placed in a queue of goals deferred for later exploration by the robotic snake. When the Turtlebot has completed exploring all reachable frontiers, it begins addressing these deferred frontiers. For each deferred goal, the ROS navigation framework is tasked to plan a feasible route up to the point where the Turtlebot is denied access; upon arriving at this location, the robotic snake is dismounted to continue traveling to the goal frontier while mapping the surrounding environment. Figure 3 depicts the overall Turtlebot exploration algorithm utilized in the marsupial robotic system architecture.

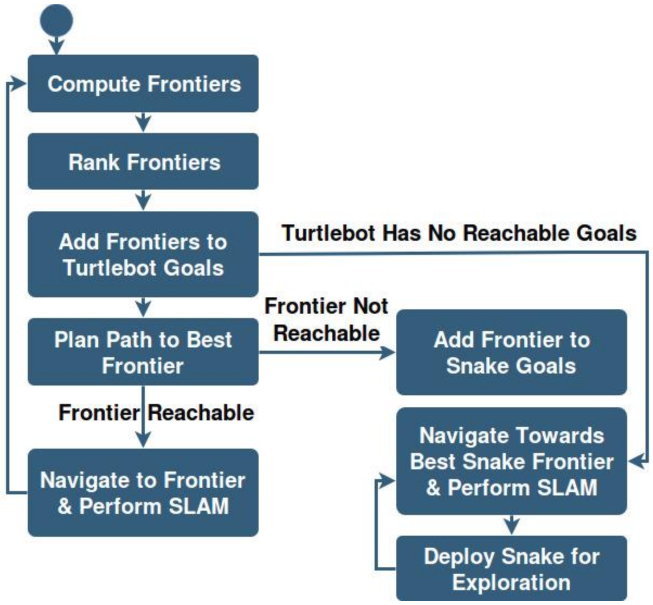

Figure 3. Block diagram of the frontier exploration architecture used by the Turtlebot.

4.2 Robotic Snake Motion Primitives

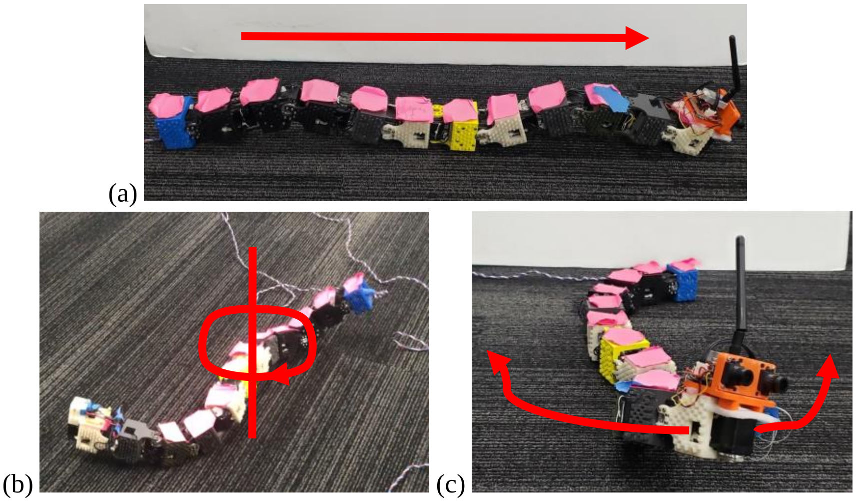

Figure 4. Robotic snake motion primitives: (a) Rectilinear gait produces forward motion, (b) Turn-In-Place gait generates in-place rotation, (c) Head Scan motion sweeps stereo camera side-to-side for widened field of view.

The robotic snake explores its environment by utilizing a series of motion primitives. These motion primitives are a set of parametrized motion plans, each with unique utilities. The traveling wave rectilinear gait (Figure 4(a)) is the primary locomotive primitive used by the robotic snake; a caterpillar-like body motion is created using a traveling sine wave to produce forward motion. Turning is accomplished by imposing a curvature along the body, parallel to the ground plane[16]. Within close proximity of obstacles or when a small radius turn is desired, the Turn-In-Place motion primitive (Figure 4(b)) is executed. Turn-In-Place rotates the robotic snake in its current position by combining distinct sinusoidal body waves defined along a horizontal plane (parallel to the ground) and along a vertical plane.

In order to localize itself within its environment, the robotic snake uses SLAM. SLAM identifies distinct features in the video feed from the stereo camera mounted on the robot's head. These features comprise points in the environment such as corners that have large pixel intensity gradients (Figure 5). As the robotic snake moves through its environment, the scene and thus the features change. By identifying corresponding features between sequential camera frames, SLAM can compute the pose change of the robot's head camera, and therefore the updated pose of the robotic snake within the environment, as it explores. Similarly to the Turtlebot, SLAM allows the robotic snake to map the world around it while, at the same time, accurately tracking its own position in that incrementally uncovered world.

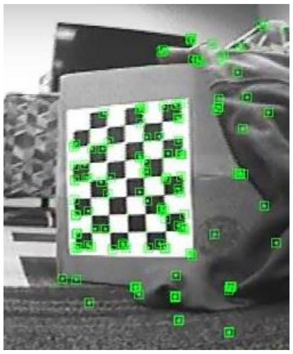

Figure 5. Output of SLAM from robotic snake stereo camera. Green rectangles indicate features within the environment recognized by SLAM.

The Head Scan motion primitive, illustrated in Figure 4(c), was developed for the robotic snake to facilitate re-localization using SLAM. When exercising the Rectilinear and Turn-in-Place gaits, unanticipated motion blur and other motion-driven artifacts in the video feed cause SLAM to operate in a degraded fashion. Instead, we permitted these gaits to run open-loop for a short period of time (e.g. 5 gait cycles), in the absence of any SLAM processing; in other words, the robot executed these gaits while effectively blind. Upon completing this short open-loop run, the Head Scan motion primitive was then executed, during which SLAM actively processed visual data from the camera. During this motion primitive, an anterior segment of the robotic snake's body, which consists of one pitch and two yaw motors, is lifted off the ground. The stereo head camera is then swept horizontally. This allowed for a slow and steady panoramic camera sweep that encompassed a widened field of view of the robotic snake's surroundings. Once SLAM features identified during a previous execution of the Head Scan motion primitive came into view again, re-localization of the robot's position in the environment was accomplished. Although the robot had travelled blindly for a short period, the subsequent execution of the Head Scan motion primitive allowed it to re-discover its location in the environment. This then facilitated the subsequent iteration of motion planning to continue exploration.

4.3 Robotic Snake Exploration Architecture

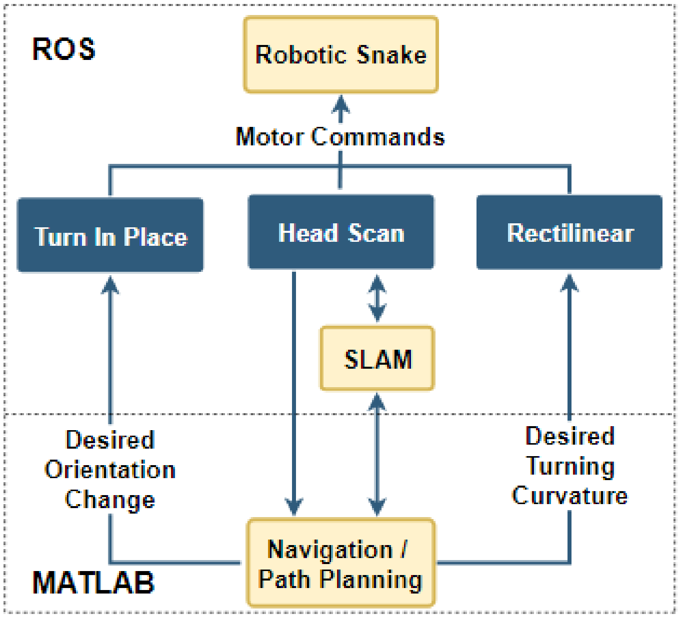

Figure 6. Block diagram of the high level exploration controller architecture used by the robotic snake.

In order to effectively transition between the different motion primitives designed for robotic snake locomotion, we developed a high level controller. The controller is developed with ROS, a middleware layer that sits between low-level robot interfaces and higher-level task and mission-oriented applications. It makes available critical infrastructure such as start-up/shutdown management, communication protocols, and shared message definitions. The controller architecture, as seen in Figure 6, describes the contributing components and flow of commands as the robotic snake explores its environment. The controller first executes the Head Scan motion primitive to localize itself in the environment. Features collected during SLAM using the stereo camera are passed to the navigation controller. The controller determines whether to execute the Rectilinear or Turn-In-Place motion primitive. After execution of one of these motion primitives the controller re-runs the Head Scan motion primitive in order to re-localize the robotic snake's pose in the environment. This process is continually repeated driving robotic snake exploration of an initially unknown environment.

5. Experimental Results

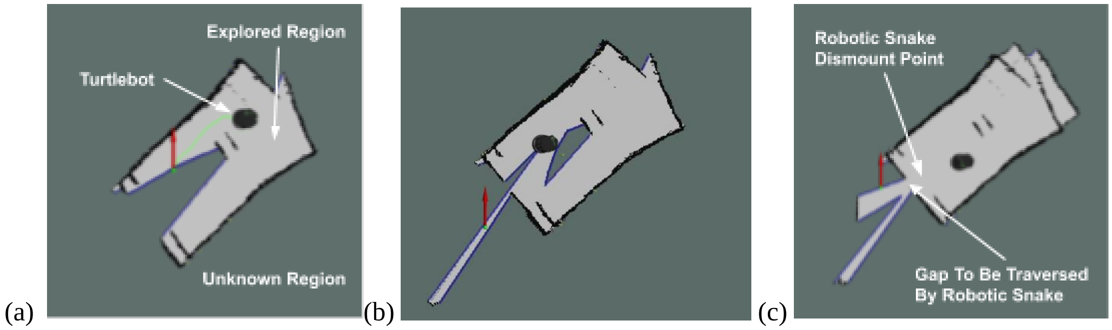

Figure 7. The Turtlebot progressively maps and explores its environment. (a) Starting configuration, (b) Turtlebot detects a new frontier, (c) Turtlebot has completed exploring reachable frontiers and mapping the environment.

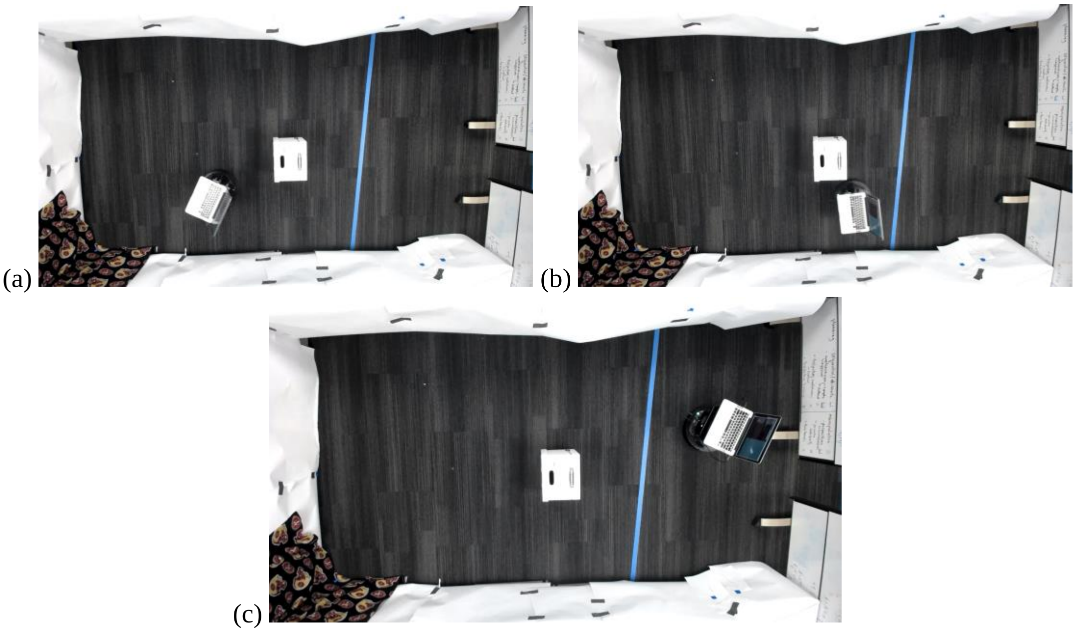

Figure 8. Physical setup of Turtlebot environment with 1-to-1 pairing to Figure 7 sub-figures. (a) Initial configuration, (b) Turtlebot explores around obstacle, (c) Turtlebot completes map and travels to robotic snake dismount point.

We demonstrate cooperative exploration of an initially unknown environment by the presented pair of mobile robots, using the developed marsupial robotic system architecture. The experimental scenario consisted of a large region of obstacle-sparse terrain connected to an obstacle-dense region; these regions were separated by a dividing barrier with a single narrow corridor allowing travel from one region to the other. Figure 8 and Figure 10 capture an overhead view of the scenario. Cabinets and chairs were used to border the entire scenario and confine the searchable area for the Turtlebot and robotic snake. Exploration began with the Turtlebot situated in a region comprising a single obstacle (Figure 8(a)). In this demonstration, a carriage and associated mounting/dismounting motion plans for the robotic snake had not yet been completed; mounting and dismounting operations were instead performed manually. The Turtlebot began exploration in the center of the region in Figure 8(a), applying frontier exploration to incrementally explore and map the reachable environment. Continual laser scans of the environment drove map construction by the Turtlebot, in Figures 7(a)-(b). The light gray region depicts explored areas and dark gray denotes unexplored areas. Blue lines depict exploration frontiers while green dots denote frontier centroids. The Turtlebot addressed the frontier list, traveling to a frontier goal characterized by the greatest utility, in Figure 8(b). This uncovered a new area of the environment; increasing the light gray area in Figure 7(b). The physical gap between the set of white boards dividing the two regions, shown in Figure 8(c), is configured to be narrower than the width of the Turtlebot and wider than the width of the robotic snake. Frontiers identified beyond this gap were classified as unreachable by the Turtlebot (Figure 7(c)). Instead, these frontiers were delegated for later exploration by the robotic snake.

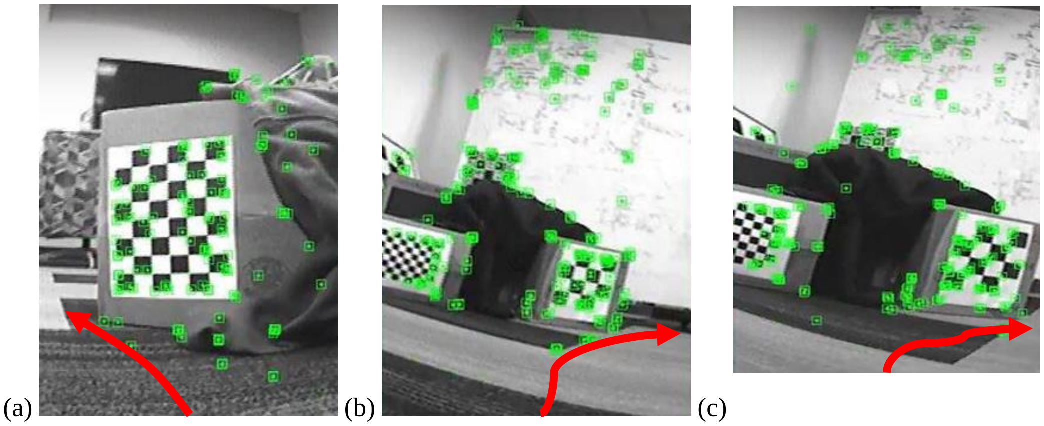

Figure 9. Output of SLAM, driven by images from stereo camera mounted on robotic snake's head. Red arrows indicate direction of motion of robotic snake. (a) Features collected of lower left obstacle in Figure 10, (b) features collected of upper right obstacle in Figure 10, (c) new features of upper right obstacle collected after movement.

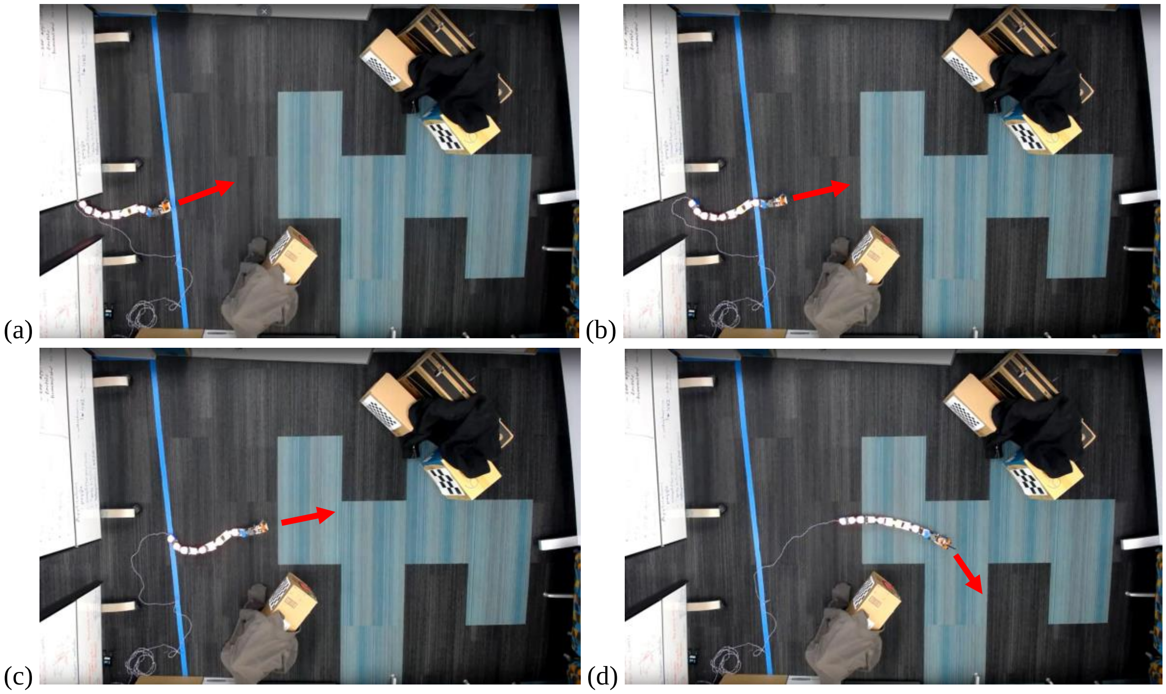

Figure 10. Robotic Snake explored and navigated its initially unknown environment. Red arrows indicate direction of motion of robotic snake. Snapshots (a)-(c) were taken to correspond with the stereo camera view in Figure 9. (a) Robotic Snake performing initial Head Scan, (b) followed by the Rectilinear gait and Head Scan, (c) Rectilinear and Head Scan sequence were then repeated, (d) turning fully around lower obstacle.

After exploring all frontiers it was capable of reaching, the Turtlebot navigated to the gap bridging the scenario regions, at which point the robotic snake was dismounted to continue exploring frontiers unreachable by the Turtlebot. This is captured in Figures 7(c) and 8(c). The robotic snake traveled through the narrow gap and continued exploration and mapping in accordance with the architecture described in Sec. 4.3. Using the Head Scan motion primitive, the robotic snake initially re-localized itself within the environment. The green squares marked in Figure 9 depict features collected and recorded during SLAM. Comparing corresponding features across different video frames enabled SLAM to ascertain how much the robotic snake had translated and rotated from frame to frame (i.e. localization). Upon completion, the robotic snake then transitioned between alternating executions of the Head Scan motion primitive and the Rectilinear gait (in an open-loop manner), in Figures 10(a)-(c). After the robotic snake completed a short open-loop (i.e. blind) execution of the Rectilinear gait, it was critical that the Head Scan motion primitive be executed next to relocalize the robot's new position in the environment. At the end of a corridor or eventual dead end (Figure 10(d)), the Turn-In-Place motion primitive would execute, rotating the robotic snake in place and preparing it for a return trip.

6. Conclusion

By pairing two mobile robot platforms, a locomotively versatile robotic snake and high-speed, high-capacity Turtlebot, the overall collaborative, multi-agent system leverages the strengths of both, while forgoing disadvantages associated with either. The Turtlebot and robotic snake comprise a cooperative marsupial robotic system better capable of exploration in obstacle cluttered environments, relative to each platform individually. An accompanying marsupial robotic architecture was developed to facilitate autonomous, cooperative exploration; using this architecture, exploratory locomotion tasks were delegated to the appropriate mobile platform based on characteristics of regions being traversed. In experiment, this marsupial pair successfully navigated and explored an initially unknown locomotive scenario, leveraging individual advantages. The Turtlebot quickly traversed largely open terrain in which a single large obstacle was situated. The robotic snake was dismounted to continue exploring areas of interest identified beyond a gap that was too narrow for the Turtlebot to fit through.

7. Future Work

In the current exploration approach, both the Turtlebot and robotic snake independently generate maps of their surrounding environments during the course of exploration. Furthermore, these maps are each represented with respect to different frames of reference. Subsequent efforts will focus on the non-trivial problem of fusing the two maps into a single global one. A common global understanding of the world becomes useful for higher-level, strategic exploration and navigation controllers that will guide the marsupial robotic pair.

Additionally, the robotic snake is not currently guided by a navigational controller; rather, it traverses the environment and simply maps it during the course of locomotion. By integrating perception-based path planning into a navigation controller, the robotic snake will gain the ability to autonomously and intelligently plan paths around obstacles and track planned paths in real-time.

8. Acknowledgements

The authors would like to thank the Opportunity Research Scholars program from the Georgia Institute of Technology's (Georgia Tech.) School of Electrical and Computer Engineering (ECE) and its sponsor, Texas Instruments. We would also like to thank Dr. Patricio Vela and Alexander Chang for their continued support in conducting this research within the Intelligent Vision and Automation Laboratory, a part of the Institute for Robotics and Intelligent Machines and School of ECE at Georgia Tech.

9. References

1. R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen, "Search and Rescue Robotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1151-1173.
2. R. R. Murphy, " Navigational and Mission Usability in Rescue Robots," *Journal of the Robotics Society of Japan*. 28. pp. 142-146.

3. F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Ø. Stavdahl, and P. Liljebäck, “A review on perception-driven obstacle-aided locomotion for snake robots,” in 2016 International Conference on Control, Automation, Robotics and Vision (ICARCV), Nov 2016, pp. 1–7.

4. Karl D. Iagnemma, Adam Rzepniewski, Steven Dubowsky, Paolo Pirjanian, Terrance L. Huntsberger and Paul S. Schenker, “Mobile robot kinematic reconfigurability for rough terrain” in Spie Digital Library, 2000.

5. R. R. Murphy, "Marsupial and shape-shifting robots for urban search and rescue," *IEEE Intelligent Systems and their Applications*, vol. 15, pp. 14-19, 2000.

6. A. Gmiterko and I. Virgala, "Simplified model of the snake rectilinear motion," 2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Smolenice, 2011, pp. 307-310. doi: 10.1109/SAMI.2011.5738895

7. H. Marvi, J. Bridges, and D. L. Hu, "Snakes mimic earthworms: propulsion using rectilinear travelling waves," Journal of The Royal Society Interface, vol. 10, 2013.

8. A. Chang and P. Vela, “Evaluation of bio-inspired scales on locomotion performance of snake-like robots,” Robotica, 2019.

9. H. Maruyama, K. Ito, “Semi-autonomous snake-like robot for search and rescue” Proceedings of 8th IEEE International Workshop on Safety, Security, and Rescue Robotics, 1-6, 2010.

10. H. Hourani, P. Wolters, E. Hauck, and S. Jeschke, "A Marsupial Relationship in Robotics: A Survey," in *Intelligent Robotics and Applications,* Berlin, Heidelberg, 2011, pp. 335-345.

11. G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A Taxonomy for Multi-Agent Robotics," in *Autonomous Robots*, vol. 3, pp. 375-397, 1996.

12. TurtleBot2, www.turtlebot.com/turtlebot2/.

13. “About ROS.” ROS.org, www.ros.org/about-ros/.

14. Y. Zhao and P. A. Vela, "Good Feature Selection for Least Squares Pose Optimization in VO/VSLAM," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1183-1189. doi: 10.1109/IROS.2018.8593641

15. G. Grisetti, C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," in IEEE Transactions on Robotics, vol. 23, no. 1, pp. 34-46, Feb. 2007. doi: 10.1109/TRO.2006.889486

16. A. Chang and P. Vela, “Closed-loop path following of traveling wave rectilinear motion through obstacle-strewn terrain,” in IEEE International Conference on Robotics and Automation, May 2017, pp. 3532–3537.