Tangible Robotic Fleet Control

AAMAS Demonstration

David St-Onge
Polytechnique Montréal
Montréal, Canada
david.st-onge@polymtl.ca

Vivek Shankar Varadharajan
Polytechnique Montréal
Montréal, Canada
vivek-shankar.varadharajan@polymtl.ca

Giovanni Beltrame
Polytechnique Montréal
Montréal, Canada
giovanni.beltrame@polymtl.ca

ABSTRACT

The use of multi-robot teams for field missions is increasing in number and scope, requiring command and control interfaces to be adapted to the operator's needs. Instead of working on interaction modalities that emerge from the engineering realm (screen, gestures or voice), we look at how the humanitarian and military logistics teams collaborate: using physical maps. In this demo, we present our command center, which consists of a swarm of small tabletop robots used to visualize and control a fleet of flying robots. To ensure the scalabilty and robustness of our control system, we leverage decentralized behaviors written in a swarm-specific programming language. We set an example scenario, where the operator must command the fleet to search an area for simulated features of interest using the tabletop robots over a map. The actions of the operator send the flying robots to individual waypoint targets. Meanwhile, the command center monitors the operator: if he is not alone, he may lack focus, and the fleet thus switches to an autonomous deployment mode until the operator's full attention is back.

KEYWORDS

UAV; tangible interfaces; Human-Robot Interaction; exploration robots

ACM Reference Format:

David St-Onge, Vivek Shankar Varadharajan, and Giovanni Beltrame. 2019. Tangible Robotic Fleet Control. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Unmanned robotic systems will soon become unavoidable for many applications unsafe to humans. Still, most commercial applications of outdoor robots are limited to the deployment of a single unit (e.g. bomb defusing, scouting), with only a handful of experimental applications with multi-robot teams [9]. In post-disaster emergency response, for instance, successful missions highlighted the importance of collaborative and complementary work between human and robots, also known as a *coactive* approach [10]. However, reports from field deployment of multi-robot teams showed that teleoperation in stressful contexts leads to mistakes [5]. To answer the call for help of humanitarians [6], we have to develop strategies decreasing the cognitive load of the operators.

In multi-robot systems, the operator's cognitive load can be reduced in two ways: 1. by giving more intelligence and autonomy to the robotic system, 2. by designing a command interface intuitive and natural to the operator. The first aspect can be addressed with robust and scalable control solutions, derived for instance from swarm intelligence. As an example, SMAVNET (EPFL) showed flying-wind UAVs to help emergency responders [4]. As with many other early demonstrations of decentralized behaviors for this application, the system was fully autonomous, designed to provide users with network communication and aerial imagery [6]. However, these implementations do not consider coactive work in human-robot missions. The system is not aware of all the information available to the responders such as which area requires more attention or where the field power station is located. Global Information Systems (GIS) have already been used for years to share information over a virtual two-dimensional map, and mission planners have leveraged this expertise for many commercial UAV systems (DroneDeploy, Pix4DCapture). This is currently the state-of-the-art for multi-robot deployment, but it requires from the operator to input commands through a tablet or a computer station, both limiting the potential for other parallel tasks to be performed by the operator.

Taking a step back, we looked at the origin of the GIS and mission planners: the physical map. Collaborative logistics plans have been successfully completed using a simple map, either pinned to a wall or lying on a table for centuries. With the help of miniature robots design, a portable localization system, and image processing, we designed a smart tangible map-based command center for exploration fleets. This paper summarizes the work done on the command center and then on the decentralized control of the fleet.

2 INTUITIVE MAP-BASED TANGIBLE COMMAND CENTER

The command center consists of a swarm of small wheeled robots mimicking the movements of the fleet over a map paired with an attention monitoring system. Figure 1 illustrates the overall architecture of the system.

2.1 Tabletop robots

The command center appears as a tent under which tabletop robots move according to the movement of a deployed robotic swarm in the field. The small robots are equipped with short-range radio communication devices and an RGB LED on the top. Each knows its position and, after initialization, each is attributed a unique aircraft counterpart on the field. Communication with the fleet passes

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

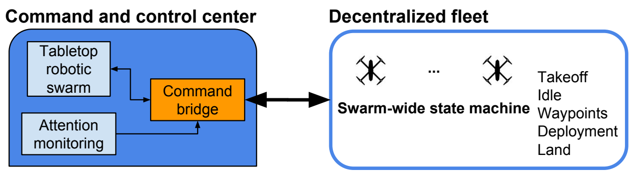

Figure 1: Overall architecture of the command center and fleet control

through a central communication node for long-range transmission. The LED color shows the aircraft battery level in real time (green, yellow or red) with different blinking patterns for different behavioral states. No blinking means the operator has full control over all the units location: it is then possible to send a waypoint command to a flying robot by holding its wheeled counterpart to the desired position for two seconds. If the wheeled robots blink, the command center detects a lack of attention from the operator and sets the fleet to autonomous deployment mode.

2.2 Attention monitoring

As mentioned above, critical missions are stressful for the operator and often require multiple simultaneous tasks to be completed. However, a highly dynamic fleet with limited flight time requires the user to stay focused for maximum performance. A number of sophisticated methods such as pupil tracking, skin conductivity, and pulse variations can be used to determine the cognitive load of the user, but these methods do not differentiate the source of the stress. Monitoring through object detection the area around the command center can provide insights on the involvement of the user.

We use a real-time object detection system (YOLOv3 [7]) feeding from a camera on the side of the command center that films the operator. The system sends periodic broadcasts to the fleet containing the objects found at the command center. The onboard behavioral script deployed on each robot accesses this object list if available and determines the number of users found in the scene. If more than one user is found, the system determines that the operator may be distracted and switches to autonomous deployment, preventing any inputs for the operator. Otherwise the system allows the operator to manipulate each unit freely, as explained above.

3 SMART DECENTRALIZED FLEET CONTROL

The development of decentralized behaviors is very challenging, especially considering that swarms are based only on local interactions with their neighbors. To accelerate the implementation of swarm behaviors, we use the Buzz programming language, which provides special constructs: shared memory (virtual stigmergy), and neighbor management. Example scripts are available online[1] as well as its runtime virtual machine, agnostic to the deployment platform[2].

With large areas to cover, limited communication range and the command center located in a safe and accessible area, it might not be always feasible to maintain a reliable connection to all the robots in the group. The decentralized control only requires at least one of the robot to be within communication range with the ground station to send commands and receive status updates over the air (Wifi mesh, Xbee, Zigbee, etc.).

3.1 Dynamic waypoints

The default mode of the command center and its fleet replicates the features of common mission planners. The waypoint commands sent to any of the UAV in the swarm from the map-based tangible interface are propagated to the fleet through any UAV within range. This UAV shares the goal among the swarm using a (key,value) pair based sharing mechanism called *virtual stigmergy*. Virtual stigmergy is a decentralized mechanism that relies on gossip based broadcasts to share data among robots in a swarm. While the control strategy is centralized, i.e. sending an absolute goal location to each robot separately, the communication leverages our decentralized system to ensure robustness.

3.2 Autonomous deployment

As mentioned above, the autonomous exploration task in swarms inspired several decentralized algorithms [8]. The approach we selected to optimize the area coverage comes neither from biological observation nor network science, but from computational geometry. The challenge is to split the region under scrutiny in smaller subregions, a process often referred to as tessellation. One algorithm that has been extensively studied for multi-robot deployment is the Voronoi tessellation [1]. It usually takes the current robot position as *seeds* to the tessellation problem and then splits the area. The logic is simple: create a frontier halfway between each two neighboring robots and then stop those lines when they cross another (or the region borders). We integrated the sweeping line algorithm in Buzz, also known as Fortune's algorithm, to extract the cell lines from a set of seeds [3]. From that point, each robot has knowledge of its cell's limits. For a uniform distribution of the robots in the area, we use a simple gradient descent toward the centroid of each cell, such as in the work of [2]. However, if the operator does not come back to its command task fast enough, each UAV generates random goals within its own cell to refine the exploration.

4 CONCLUSION

On top of addressing the need for intuitive command interfaces for humanitarians, this design will help other remote operations, by granting the developer with a control of the perceived latency between the real deployed robots and the ones from the command center. Our implementation can be adapted for underwater or underground exploration, and even planetary exploration. This demonstration will provide participants with a real experience of such a multi-robot remote exploration mission. A draft of the demonstration video is available online: https://youtu.be/d_aqJB_0smU.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of NSERC funding (STPGP 479149).

[1] http://the.swarming.buzz/ICRA2017/cheat-sheet/
[2] https://github.com/MISTLab/Buzz

REFERENCES

[1] Vladimir Alexandrov, Konstantin Kirik, and Alexander Kobrin. 2018. Multi-robot Voronoi tessellation based area partitioning algorithm study. *Journal of Behavioral Robotics* (2018), 214–220.

[2] Jorge Cort, Member Ieee, Sonia Mart, Member Ieee, Timur Karatas, Francesco Bullo, Member Ieee, Jorge Cortés, Sonia Martínez, Timur KarataÅ§, and Francesco Bullo. 2004. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation* 20, 2 (2004), 243–255. https://doi.org/10.1109/TRA.2004.824698 arXiv:math/0212212

[3] Steven Fortune. 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2, 1 (01 Nov 1987), 153. https://doi.org/10.1007/BF01840357

[4] Sabine Hauert, Severin Leven, Maja Varga, Fabio Ruini, Angelo Cangelosi, Jean Christophe Zufferey, and Dario Floreano. 2011. Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate. *IEEE International Conference on Intelligent Robots and Systems* (2011), 5015–5020. https://doi.org/10.1109/IROS.2011.6048729

[5] Geert-jan M Kruijff, Viatcheslav Tretyakov, Thorsten Linder, Sankt Augustin, Fiora Pirri, Mario Gianni, Panagiotis Papadakis, Matia Pizzoli, Arnab Sinha, Emanuele Pianese, Salvatore Corrao, Fabrizio Priori, Sergio Febrini, Sandro Angeletti, and Vigili Fuoco. 2012. Rescue Robots at Earthquake-Hit Mirandola , Italy : a Field Report. In *International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 1–8. https://doi.org/10.1109/SSRR.2012.6523866

[6] Mark Lliffe. 2016. *Drones in Humanitarian Action*. Technical Report. The Swiss Foundation for Mine Action, Geneva/Brussels. http://drones.fsd.ch/

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Madhubhashi Senanayake, Ilankaikone Senthooran, Jan Carlo Barca, Hoam Chung, Joarder Kamruzzaman, and Manzur Murshed. 2016. Search and tracking algorithms for swarms of robots: A survey. *Robotics and Autonomous Systems* 75 (2016), 422–434. https://doi.org/10.1016/j.robot.2015.08.010

[9] Bruno Siciliano and Oussama Khatib (Eds.). 2018. *Handbook of Robotics (2nd Edition)*. Springer.

[10] Daniel Szafir, Bilge Mutlu, and Terrence Fong. 2017. Designing planning and control interfaces to support user collaboration with flying robots. *International Journal of Robotics Research* (2017), 1–29. https://doi.org/10.1177/0278364916688256