UAV Teams In Emergency Scenarios: A Summary Of The Work Within The Project PRISMA

Carmine Recchiuto, Antonio Sgorbissa, Francesco Wanderlingh, and Renato Zaccaria

DIBRIS Department, University of Genova, via all'Opera Pia 13, 16145, Genova, Italy
`carmine.recchiuto@dibris.unige.it`

Abstract. In recent years autonomous robots, and Unmanned Aerial Vehicles (UAVs) in particular, are becoming always more important in the context of emergency scenarios, being able to anticipate the actions of human operators and to support them during rescue operations. In this context, the investigation of strategies for the autonomous control of UAVs, for the development of Human-Swarm Interfaces and for the coverage of large areas is crucial. All these aspects have been analyzed within the Italian project PRISMA, and they will be here summarized.

Keywords: UAVs, monitoring, Search&Rescue, Human-Swarm Interfaces, coverage algorithms, virtual reality

1 Introduction

The work described in this article has been performed during the PRISMA project, which focusses on the development and deployment of robots and autonomous systems able to operate in emergency scenarios, with a specific reference to monitoring, pre-operative management, and real-time intervention. The work has been focussed on Unmanned Aerial Vehicles (UAVs), being able of monitoring a wide area in a small time, quickly moving and being easily controlled by human operators. In particular, some aspects have been analyzed more in details: techniques for localization and autonomous control, coverage algorithms, strategies for moving in a structured formation and the integration of virtual reality tools for visualization and control.

2 Indoor localization and autonomous control

An indoor experimental setup can be extremely useful when dealing with aerial robots, in order to speed up the development of model and algorithms. In this context, the main problem is related to the localization of the robots, since GPS signal is denied. In the project the problem was solved using a camera (MatrixVision, mvBlueFox) on board of the esarotor Asctec Firefly [1] and integrating the artificial vision library ArUco [2]. The principal functionality of the library is to recognize up to 1024 different markers, applying an Adaptive Thresholding and

Fig. 1. The markers used for localization and the esarotor Asctec Firefly hovering

the Otsu's algorithm [3]. When a marker is recognized, the relative distance and orientation of the camera with respect of the marker is given.

For improving the accuracy of the localization, a wall of 35 markers has been created (Fig. 1), and a custom algorithm has been developed, based on the elimination of the outliers and the estimation of the average value. The reference in position is then used for the actual control of the robot. Indeed, the UAV calculates the error between the target position (a fixed value when hovering, a series of waypoints in more complex cases) and uses the error in space as input of three PID controllers for the three directions in space. The resulting target accelerations are used to calculate the reference thrust u and the control angles ϕ_d (pitch) and θ_d (roll), considering the dynamics of the system, the mass m of the UAV and the angle ψ_d (yaw):

$$\begin{aligned} u &= m\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2} \\ \phi_d &= \sin^{-1}\left(m\frac{\mu_x \sin\psi_d - \mu_y \cos\psi_d}{u}\right) \\ \theta_d &= \tan^{-1}\left(\frac{\mu_x \cos\psi_d - \mu_y \sin\psi_d}{\mu_z + g}\right) \end{aligned}$$

The control of the orientation of the multirotor (ψ_d) has been achieved with a proportional controller, directly based on the error between the reference angle and the actual one. The whole control software, composed of the localization system, the PID controllers and other modules dedicated to the planning of the actions (i.e., taking-off, hovering, reaching a waypoint, taking a picture, landing) and to the interfacing with the user, has been implemented on board of the esarotor Asctec Firefly, within the ETHNOS framework [4], a programming environment for the design of real time control systems.

3 Coverage algorithms for Search & Rescue

With a similar control approach, but using mainly GPS as localization system, a monitoring strategy has been implemented and tested outdoor, using two multirotors (Asctec Pelican and Asctec Firefly). The main idea was to analyze and compare the performances of some *real-time* multi-robot coverage algorithms (i.e., *Node Count*, *Edge Counting*, *Learning Real-Time A** and *PatrolGRAPH**) [5] [6], aimed at finding a decision procedure allowing a team of robots to navigate in a workspace modelled as a navigation graph.

The four algorithms have been firstly tested in simulation to the end of comparing their performances, considering as indicators the length of the longest path among all the robots and the overall distance travelled by all robots. The results confirmed many of the previous works in literature, suggesting in particular that the *Node Count* algorithm, despite its simplicity, is the most efficient one. This becomes more evident increasing the number of robots and the size of the grids, mainly because of the update rule of other high-performance algorithms (e.g. LRTA*).

Finally, the algorithms have been practically implemented using the two multirotors, in order to test the whole framework (Fig. 2). A ROS/ETHNOS interface has been developed to implement the communication between the off-board controller (executing the algorithms) and the two robots.

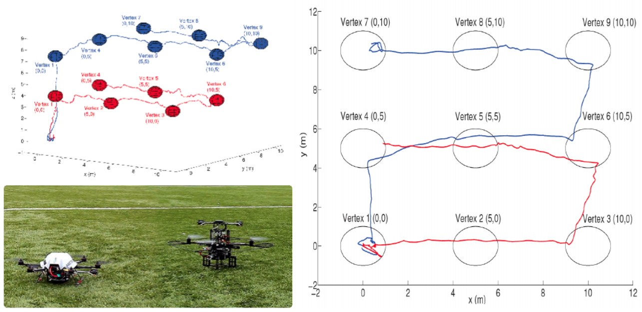

Fig. 2. The two multirotors and an example of the paths followed using a 3x3 grid and the Node Count algorithm.

4 Movement in formation and implementation of a custom Human-Swarm Interface

Even if many steps forward have been taken towards the fully autonomous control of UAVs, a human pilot is usually in charge of controlling the robots. However, teleoperating UAVs can become a hard task whenever it is necessary to deploy a swarm of robots instead of a single unit, to the end of increasing the area under observation. In this case, the organization of robots in a structured formation may reduce the effort of the operator.

For all these reasons, a custom Human-Swarm Interface has been built, allowing human operators to control a team of multirotors in environments filled with obstacles. The algorithm is mainly based on the work of Balch and Arkin

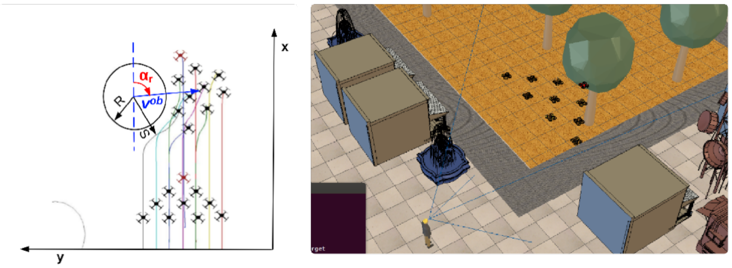

Fig. 3. On the left, a wedge formation avoiding a circular obstacle. On the right, the simulated environment for the experimental phase.

[7], with a unit-center approach and the organization of the whole strategy as a sum of concurrent behaviours (i.e., *avoiding obstacles*, *avoiding inter-robots collision*, *following user commands*, *keeping the formation*, in a descending priority order), handling a certain number of predetermined typologies of formation, and receiving user inputs by means of a two-axis joypad (Fig. 3). The HSI has been tested with a simulated environment, investigating also the effect of different point of views on the user performances, showing a strong relation between human performances, typology of the task and situational awareness. In particular, it has been shown that a first person point of view is suitable for some typologies of tasks, where a direct view of the environment is sufficient, whereas a more evident degradation of the performances is noticed in a task where a higher level of situational awareness is necessary.

5 Integration of a virtual reality platform

Given the necessity of easing the control of the robot from the operator point of view, the integration of virtual reality tools has also been investigated. In particular, the Oculus Rift [8], a virtual reality head-mounted display, has been used in order to give inputs to the robot (or to the whole swarm in simulation) and to visualize the images taken from the on-board cameras (Fig. 4).

More in details, the inertial sensors embedded in the head-mounted display are used to periodically measure the yaw orientation of the user's head, using it as reference for the yaw control of the multirotor. The ROS/ETHNOS bridge has again been used for implementing the bidirectional communication (angles and video streaming).

Fig. 4. Oculus Rift (left) and related images taken with on-board cameras and in simulation.

References

1. Achtelik, M. C., Doth, K. M., Gurdan, D., and Stumpf, J. (2012). Design of a Multi Rotor MAV with regard to Efficiency, Dynamics and Redundancy. In AIAA Guidance, Navigation, and Control Conference (pp. 1-17).
2. Munoz-Salinas, R. (2012). ARUCO: a minimal library for Augmented Reality applications based on OpenCv.
3. Hao, Y. M., and Zhu, F. (2005). Fast Algorithm for Two-dimensional Otsu Adaptive Threshold Algorithm [J]. Journal of Image and Graphics, 4, 014
4. Piaggio, M., Sgorbissa, A., and Zaccaria, R. (2000). A programming environment for real-time control of distributed multiple robotic systems. Advanced Robotics, 14(1), 75-86.
5. Koenig, Sven, Boleslaw Szymanski, and Yaxin Liu. "Efficient and inefficient ant coverage methods." Annals of Mathematics and Artificial Intelligence 31.1-4 (2001): 41-76.
6. Baglietto, M., Cannata, G., Capezio, F., Grosso, A., Sgorbissa, A., and Zaccaria, R. (2008, July). PatrolGRAPH: a distributed algorithm for multi-robot patrolling. In IAS10-The 10th International Conference on Intelligent Autonomous Systems, Baden Baden, Germany (pp. 415-424).
7. Balch, T., and Arkin, R. C. (1998). Behavior-based formation control for multirobot teams. Robotics and Automation, IEEE Transactions on, 14(6), 926-939.
8. Oculus, V. R. (2015). Oculus Rift. Available from: http://www.oculusvr.com/rift.