The project PRISMA: Post-Disaster Assessment with UAVs

Conference Paper *in* Advances in Intelligent Systems and Computing · July 2016

DOI: 10.1007/978-3-319-48036-7_15

2 authors:

Carmine Tommaso Recchiuto
Università degli Studi di Genova
50 PUBLICATIONS **223** CITATIONS

SEE PROFILE

Antonio Sgorbissa
Università degli Studi di Genova
168 PUBLICATIONS **1,247** CITATIONS

SEE PROFILE

The project PRISMA: Post-Disaster Assessment with UAVs

Carmine Tommaso Recchiuto[1] and Antonio Sgorbissa[1]

Università degli studi di Genova, DIBRIS Department,
via all'Opera Pia 13, 16145, Genova, Italy
carmine.recchiuto@dibris.unige.it

Abstract. In the context of emergency scenarios, Unmanned Aerial Vehicles (UAVs) are extremely important instruments, in particular during monitoring tasks and in relation to the Post-Disaster assessment phase. The current paper describes a summary of the work performed during PRISMA [1], a project focused on the development and deployment of robots and autonomous systems able to operate in emergency scenarios, with a specific reference to monitoring and real-time intervention. Among other aspects, the investigation of strategies for mapping and for path following, for the implementation of Human-Swarm Interfaces and for the coverage of large areas have been performed, and they will be here summarized.

Keywords: UAVs, Multirotors, Post-Disaster Assessment, Image Stitching, Path Following, Coverage Algorithms, Human-Swarm Interfaces

1 Introduction

In recent years, the area of Unmanned Aerial Vehicles (UAVs) research has shown rapid developments, giving the possibility of deploying these typology of robots in a wide range of domains: agriculture, environmental monitoring, sport events, cinema, Search&Rescue and emergency scenarios.

The usage of UAVs in the Search&Rescue (SAR) domain has widely been investigated: indeed, small aerial vehicles give the possibility to quickly monitor large areas, with a high maneuverability while moving in environments filled with debris and obstacles. For all these reasons, many research teams have already practically tested the deployment of aerial robots in disaster scenarios: the Pratt team [2] implemented small UAVs after the Hurricane Katrina, for inspecting buildings in urban environments, performing omni-directional obstacle avoidance. In the same years, the work of Murphy [3] [4] focused on the analysis of a cooperative strategy between ground vehicles and UAVs, for seawall and bridge inspections. UAVs were also deployed in the context of the L'Aquila earthquake [5] and of the Lushan earthquake [6].

The EU founded projects ICARUS and NIFTi are also worth of mention. The main aim of the project ICARUS [7] was the improvement of crisis management with the use of unmanned SAR robotic systems embedded and integrated into existing infrastructures. The aerial platforms included also solar rotor-wing and fixed-wing UAVs, developed by ETH for sustained flight [8]. The NIFTi project [9] focused on human-robot cooperation, and, in the context of the project, a novel octocopter equipped with a camera and a Kinect-style sensor was designed and developed.

As evident from all these works, the typologies of UAVs adopted in these scenarios are different: however, it can be seen that, while in the earliest approaches the preferred choice was small fixed-wing UAVs and miniature helicopters [4], [10], multirotors have been used in more recent events, given their limited cost and the ability of vertically taking-off and landing [7], [11] .

This work started from all these basis, analyzing the principal issues for SAR performed with UAVs (in particular in the Post-Disaster Assessment phase) and proposing some solutions to common problems. This paper summarizes the work performed in the project PRISMA, that focused in particular on the integration of the ETHNOS real-time framework, on the implementation of novel strategies for path following, on the analysis of the performances of a subset of coverage algorithms, and on tests with human subjects aimed at studying Human-Swarm Interfaces. The following paragraphs will describe all these aspects.

2 The Asctec Multirotors and the ETHNOS Framework

As stated in the Introduction, multirotors are nowadays a common choice in works involving UAVs, mainly because of their maneuverability. In the context of the project PRISMA, two multirotors have been used: the Asctec Firefly (esarotor) and Pelican (quadrotor) [12] (Fig. 1).

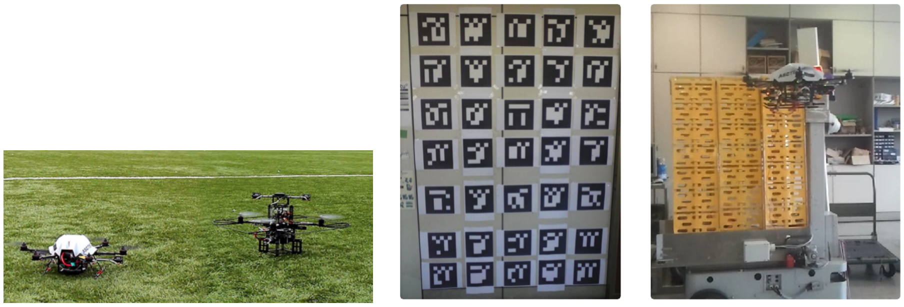

Fig. 1. The Asctec Firefly and Pelican during an experimental session (left). The Asctec Firefly during an indoor flight, localized with a board of markers (right).

The two platform are endowed with a Flight Control Unit (FCU) *Autopilot*. The FCU features a complete Inertial Measurement Unit (IMU) as well as two 32 bits, 60 Mhz ARM-7 microcontrollers used for flight control and data fusion. The two microcontrollers have different aims: the first, the Low-Level Processor (LLP) is responsible for the IMU sensor data fusion and for the attitude-based controller, while the second, the High-Level Processor (HLP) may be used to implement further custom code. For the computationally more expensive on-board processing tasks (e.g., image processing for localization) there is also an embedded computer, with a Intel Core I7 CPU, a 1 GB RAM, a MicroSD card slot for the operating system.

In the proposed implementation, the HLP was considered as a bridge for the serial communication between the LLP and the on-board CPU, where the control code was implemented (Fig. 2). For the control development and for the communication between robots and human operators, the ETHNOS framework [13] was adopted.

ETHNOS (Expert Tribe in a Hybrid Network Operating System) is a programming environment for the design of real-time control systems. From the runtime perspective, ETHNOS provides the possibility of real-time executing periodic and sporadic tasks, a feature that is crucial in aerial vehicles. Indeed, a dedicated network communication protocol, within the distributed operating system, allows a robust communication even in noisy wireless network. The framework achieves soft real-time compliance being a real-time extension to the standard Linux kernel [14]. The reference architecture of the ETHNOS operating system is based on the concept of *expert*, a concurrent agent responsible for a specific deliberative or reactive behavior. The environment handles the concurrent execution and the scheduling of the experts, and their synchronization when accessing the shared resources.

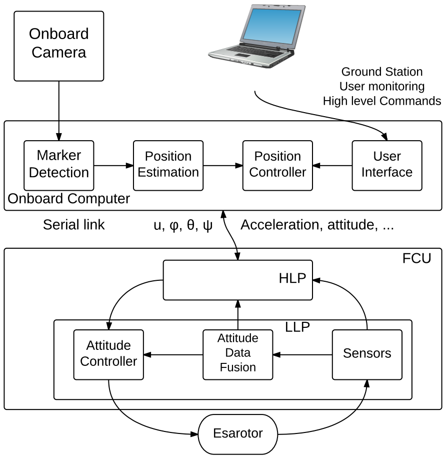

Fig. 2. Architecture of the whole system

An example of the ETHNOS architecture is the experimental setup developed for indoor localization and autonomous control. The control and indoor localization of the Asctec multirotors have been performed on board using the concurrent structure based on the concept of expert. In this case, the control of the robot has been implemented using five periodic experts:

- *Acquisition*: handles the communication with the HLP (and consequently with the LLP) with a frequency of 100 Hz, periodically reading the data coming from the on-board sensor and sending to the FCU the commands related to thrust and attitude angles, calculated by the expert Control (see below).
- *Localization*: implements an algorithm for indoor localization, based on the artificial vision library ArUco [15]. The principal functionality of the library is to recognize up to 1024 different markers, applying an Adaptive Thresholding and the Otsu's algorithm. When a marker is recognized, the relative distance and orientation of the camera with respect of the marker is calculated. For improving the accuracy of the localization, a wall of 35 markers has been created (Fig. 1, right) and a custom algorithm has been developed, based on the elimination of the outliers and the estimation of the average value. The expert operates at a frequency of 30Hz.
- *Interface*: works at 5 Hz, and it offers a simple User Interface integrated in the Unix Shell that allows the user to select the operation to be performed
- *Planner*: acts as a simple bridge between Interface and Control, in case of complex actions (i.e. waypoint too far, list of commands to be executed), with a frequency of 10Hz.
- *Control*: sporadic expert, it uses the information deriving from the other experts (i.e. position, attitude, target position, ...) for the control of the robot. Actually, 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 direction in space:

$$\begin{aligned}
u_x &= K_{Px}(x - x_T) + K_{Dx}\tfrac{d}{dt}(x - x_T) + K_{Ix}\textstyle\int(x - x_T) \\
u_y &= K_{Py}(y - y_T) + K_{Dy}\tfrac{d}{dt}(y - y_T) + K_{Iy}\textstyle\int(y - y_T) \\
u_z &= K_{Pz}(z - z_T) + K_{Dz}\tfrac{d}{dt}(z - z_T) + K_{Iz}\textstyle\int(z - z_T))
\end{aligned}$$

where the K coefficients are the proportional, derivative and integrative proportional terms and x_T, y_T and z_T are the references in position. Starting from the resulting target accelerations u_x, u_y and u_z, these values 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.

$$\psi_d = K_{P_\psi}(\psi - \psi_T)$$

Finally, u, ϕ_d θ_d and ψ_d are used as inputs by a inner control loop, implemented in the LLP, that directly handles the rotor velocities.

The real-time scheduling functionalities of ETHNOS are crucial for vertex-to-vertex navigation, achieved as shown for the indoor case. However, given the widespread diffusion of the framework ROS (Robot Operating System), this robotic middleware has also been integrated for some applications; therefore, within the project, a ROS-ETHNOS interface has been developed. The interface is able to convert ROS messages into ETHNOS ones, and vice versa, e.g. converting position commands from ROS topics to ETHNOS messages. An example of this integration will be presented in section 5.

3 SLAM & Image Stitching

Dealing with aerial robots in unknown scenarios, localization and map building are tasks of the utmost importance. While commonly used SLAM techniques are based on Extended Kalman Filters [16] or particle filters [17], the approach used in the project PRISMA focused on the implementation of a Image Stitching algorithm. A detailed description of this typology of algorithms can be found in [18], while in this section, we will just focus on more technical aspects related to the practical implementation.

In details, the practical implementation of the algorithm has seen the adoption of feature-based detection algorithms, such as SIFT and ORB (at the discretion of the user) that allow reducing computational costs and incrementing execution speed [19]. Both algorithms start from a scan of the selected images, in order to create the descriptors with the related features; after this step, the features are evaluated in subsequent images, and if the number of similar features is too small and below a certain threshold, the images are discarded. In this phase there is also the estimation of the camera parameters: if the intrinsic ones can be considered as a priori known, the extrinsic ones should be evaluated in relation to the received images.

For the Image Alignment process, the algorithm has seen the implementation of Bundle Adjustment techniques (i.e., a photogrammetry technique aimed at combining multiple images related to the same scene) and Feathering Image Blending (i.e., a blending technique based on the softening and on the blurring of the edges of the features) for the final Blending phase. Moreover, the Image Alignment process allows estimating the multirotor position, considering the altitude of the robot and the rototranslation vector of an image with respect of the previous one, strictly connected with the movement of the UAV. The whole code is made available, as a ROS package, at the following link: *https://github.com/CDibris/aerialSLAM*

The images of the two cameras, and in particular the one directed toward the ground, are streamed by means of ROS topics. The algorithm periodically samples the video stream; each time a new image is saved, the ROS nodes activates a second process, that implements the Image Stitching algorithm. Preliminary tests have been implemented, using first some video datasets available online (Fig. 3) and also directly using the on-board cameras.

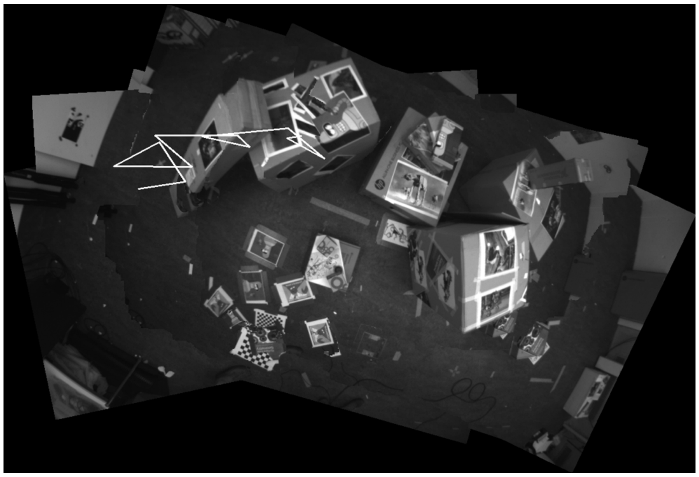

Fig. 3. Image Stitching process during a rototranslation of the robot. Dataset taken from http://rpg.ifi.uzh.ch/software_datasets.html

4 Real-time path generation with obstacle avoidance

Another fundamental aspect when speaking of SAR with UAVs is related to the ability of the robots of following predefined exploration paths, being also able to modify and update such paths in real-time, based on how they perceive the environment (i.e., recognizing and avoiding obstacles on the path). During the project, a novel algorithm for path generation and obstacle avoidance, using an uncommon technique to describe and modify the path, has been proposed and implemented on the Asctec multirotors.

The algorithm (more details can be found in [20] [21]) is based on:

- *3D Path definition*, through the intersection of two surfaces represented by implicit equations in the form $f_i(x, y, z) = 0$, $i = 1, 2$. The path is given by the solution of the system:

$$\begin{cases} f_1(x, y, z) = 0 \\ f_2(x, y, z) = 0. \end{cases}$$

- *Path modification in presence of obstacles*, i.e. when a robot perceives the presence of any obstacle, the path is deformed, by means of obstacle functions added to one of the surface functions. The system becomes:

$$\begin{cases} f_1'(x,y,z) = f_1(x,y,z) + \sum_{j=1}^{N} O_j(x,y,z) = 0 \\ f_2(x,y,z) = 0. \end{cases}$$

where a Gaussian function is used as obstacle function:

$$O_j(x,y,z) = A_j e^{-\frac{(x-x_j)^2+(y-y_j)^2+(z-z_j)^2}{\sigma^2}}$$

It should be noticed that, by computing the tuning parameters A_j and σ properly, it is possible to guarantee that the deformed path does not collide with obstacles, even in the presence of multiple ones [22]

– *Path generation*, the definition of a velocity vector that should be used as input of the quadrotor, as:

$$V_k(x,y) = -k_g f(x,y) \nabla f(x,y) + tangent(x,y)$$

where k_g is the gradient vector gain, the product $-f(x,y)\nabla f(x,y)$ drives the robot towards the desired path and $tangent(x,y)$ is a vector perpendicular to $\nabla f(x,y)$ that drives the robot along the path.

An example of the simulated behaviour and of the real execution of the algorithm with the Asctec Firefly can be seen in Fig. 4.

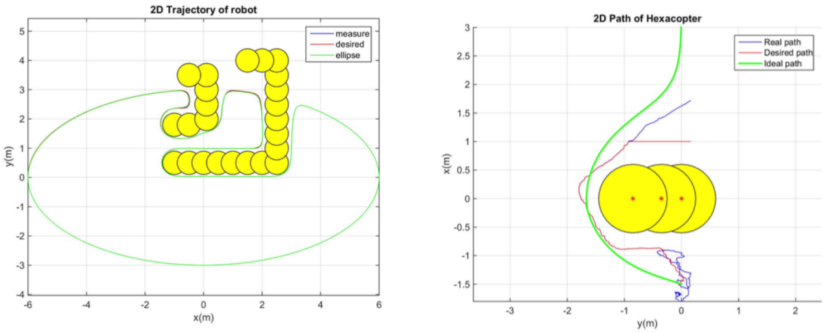

Fig. 4. Simulation of a multicopter following an ellipse with 2 nested horseshoes of obstacles (left). Flight of the Asctec Firefly in a straight line with 3 obstacles (right).

5 Coverage Algorithms for Search & Rescue

Multirotor UAVs are an extremely interesting technology towards the end of exploring an area and gathering information before first responders come into play. During the project PRISMA, we investigated different strategies for the organization of a team of UAVs implementing autonomous coverage strategies for real-time intervention after earthquakes.

The work was mainly focused in the analysis and comparison of a subset of real-time multi-robot coverage algorithms, i.e. *Node Count* [23], *Edge Counting*

[24], *Learning Real-Time A** [23], *PatrolGRAPH** [25], chosen among the most common coverage algorithms in literature, in order to classify them in term of the required coverage time and of the required energy.

For a formal description of the algorithms we refer the readers to the related literature works. In this section we will just recap the most relevant results obtained from tests in simulation, considering as indicators the length of the longest path among all robots and the overall distance travelled by all robots:

- the *Node Count* algorithm, despite its simplicity, is the most efficient one, in particular increasing the map size and the number of robots;
- the performance of the *Edge Counting* and *PatrolGRAPH** algorithms are noticeably lower;
- however, when a higher minimum number of visits for each node is considered, *PatrolGRAPH** performs better, since it ensures a uniform frequency of visits to all vertices [26].

It is worth of mention that the algorithms were also implemented on actual robots, testing the whole framework in a real scenario. In the current implementation, based on the ROS framework (the software is made available as a ROS package at the following link: *https://github.com/merosss/VRepRosQuadSwarm*), a central station is in constant communication with each robot, that has however a local copy of the navigation graph. In case of a temporary lack of global communication, different robots can continue to work, possibly with decreased performance, by using their local copies. The ROS framework and the ETHNOS architecture are integrated using the ROS/ETHNOS interface presented in section 2.

6 Movement in formation and Human-Swarm Interfaces

Even if many steps forward have been taken towards the fully autonomous control of UAVs, a human pilot is usually in charge of teleoperating the robots during SAR activities. In this context, a swarm of robots is usually deployed, and therefore the teleoperation of UAVs can become a hard task. In order to reduce the effort of the human operator, the organization of the robots in a structured formation allows to:

- increase the field of view, by positioning the robots in a way that a greater section of the surrounding environment can be visualized;
- speed up the learning process of the human operator, that is able to predict the robot motions in a more straightforward way and, consequently, to move the swarm faster;
- avoid collisions between robots;

In relation to the effort of the human operator, also the typology of visual feedback is crucial: different aspects related to the monitoring of the surrounding environment should be taken into account (e.g., the positions of the cameras, the necessity of implementing an egocentric, an exocentric view, or a hybrid combination of both, the quality of the images in relation to the available bandwidth).

In order to investigate the effects of different visual feedbacks on the human operators performances, a custom Human-Swarm Interface (HSI) has been built during the project. The HSI allows human operators to pilot a team of multirotors in environments filled with obstacles, while the robots keep a structured formation, with a behavior-based policy where the global movement of each robot is calculated as the sum of single concurrent behaviours [27]. More details about the design of the interface can be found in [28].

The interface allows the user to control the swarm both with a camera from above the swarm (exocentric point of view) and with a combination of cameras on board the robots (egocentric point of view). While it is known that an exocentric camera allows to better estimate distances and perceive the environment [29], it is also a matter of fact that, in real context, the common way to pilot quadrotors (when a direct line of sight is not possible) is by using the feedback of onboard cameras. Therefore, a series of experiments was aimed at investigating the degradation of performances with the camera in a *first person* configuration. For the aim of the experiments, an environment filled with obstacle was created, as in Fig. 5.

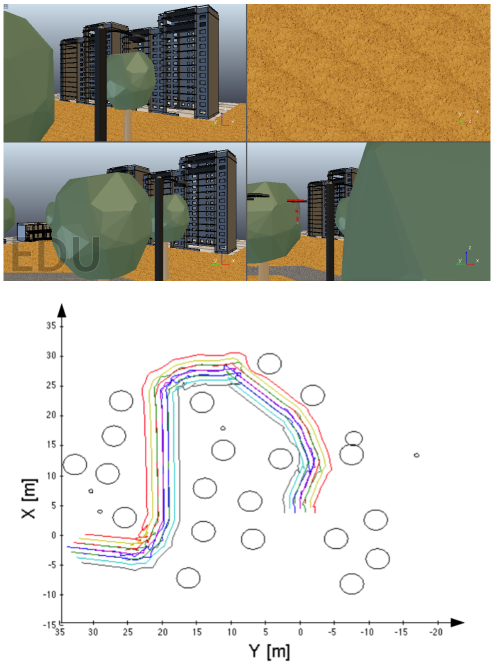

Fig. 5. Swarm in the simulated environment, with an egocentric point of view composed of two cameras on board the robot leader(pointing forward and downward) and on board the *wings* of the formation (the leftmost and rightmost robots) (up). Paths followed by the robots in a 10-UAVs wedge formation (down).

The results show that, as expected, an exocentric point of view allows to generally achieve better performances. However, it can be noticed that when the user is required to accomplish a simple task, that does not require a high situational awareness, the differences between the performances achieved with the *first person* point of view and with the camera from above were not so relevant. On the opposite, during a task consisting in manually avoiding the obstacles (Fig. 5, down), and therefore requiring a higher situational awareness, the gap in performance between the two points of view was much higher.

A more detailed discussion on the results of the experiments can be found in [28]. The experimental environment and the control software is made available, as a ROS package, at the following link: *https://github.com/CDibris/Quad Swarm.*

7 Conclusions

Unmanned Aerial Vehicles and multirotors in particular are receiving in last years a tremendous attention both by the scientific community and by the market, due to their peculiar characteristics: vertical take-off and landing, limited cost, ease of construction, flexibility. For these reasons, and given also the rapid development of UAV technology, in the last decade the scientific community has devoted much interest in the implementation of UAVs swarm in the context of Search&Rescue, and in particular in the Post-Disaster Assessment phase.

This paper presents the work performed in the context of the project PRISMA, related to the development and deployment of aerial robots in emergency scenarios, with a specific reference to monitoring and real-time intervention.

The work has taken into account different aspects related to the utilization of UAVs in an emergency context: multi-robot software architectures, SLAM techniques, path following strategies, analysis of real-time coverage algorithms, development of novel Human-Swarm Interfaces.

The deployment of autonomous aerial robots in emergency situations still presents many complex challenges and open issues. The project PRISMA aims to respond to some of these challenges, by providing some practical solutions, innovative strategies and experimental analysis, using both simulation environments and actual robots.

References

1. Recchiuto, C., Sgorbissa, A., Wanderlingh, F., & Zaccaria, R. UAV Teams In Emergency Scenarios: A Summary Of The Work Within The Project PRISMA.
2. Pratt, K. S., Murphy, R., Stover, S., & Griffin, C. (2009). CONOPS and autonomy recommendations for VTOL small unmanned aerial system based on Hurricane Katrina operations. Journal of Field Robotics, 26(8), 636-650.
3. Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., & Erkmen, A. M. (2008). Search and rescue robotics. In Springer Handbook of Robotics (pp. 1151-1173). Springer Berlin Heidelberg.

4. Murphy, R. R., Steimle, E., Griffin, C., Cullins, C., Hall, M., & Pratt, K. (2008). Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. Journal of Field Robotics, 25(3), 164-180.
5. Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., & Rinner, B. (2010). Networked UAVs as aerial sensor network for disaster management applications. e & i Elektrotechnik und Informationstechnik, 127(3), 56-63.
6. Qi, J., Song, D., Shang, H., Wang, N., Hua, C., Wu, C., ... & Han, J. (2015). Search and Rescue Rotary-Wing UAV and Its Application to the Lushan Ms 7.0 Earthquake. Journal of Field Robotics.
7. Govindaraj, S., Chintamani, K., Gancet, J., Letier, P., van Lierde, B., Nevatia, Y., ... & Armbrust, C. (2013, October). The icarus project-command, control and intelligence (c2i). In Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on (pp. 1-4). IEEE.
8. Leutenegger, S., Jabas, M., & Siegwart, R. Y. (2011). Solar airplane conceptual design and performance estimation. Journal of Intelligent & Robotic Systems, 61(1-4), 545-561.
9. Kruijff, G. J. M., Kruijff-Korbayov, I., Keshavdas, S., Larochelle, B., Jan?ek, M., Colas, F., ... & Looije, R. (2014). Designing, developing, and deploying systems to support human?robot teams in disaster response. Advanced Robotics, 28(23), 1547-1570.
10. Murphy, R. R. (2014). Disaster robotics. MIT press.
11. Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., ... & Ohno, K. (2012). Collaborative mapping of an earthquake?damaged building via ground and aerial robots. Journal of Field Robotics, 29(5), 832-841.
12. Ascending Technologies GmbH, website, http://www.asctec.de.
13. Piaggio, M., Sgorbissa, A., & Zaccaria, R. (2000). A programming environment for real-time control of distributed multiple robotic systems. Advanced Robotics, 14(1), 75-86.
14. Sgorbissa, A. (2006). Multi-Robot Systems and Distributed Intelligence: The ETHNOS Approach to Heterogeneity. INTECH Open Access Publisher.
15. Munoz-Salinas, R. (2012). ARUCO: a minimal library for Augmented Reality applications based on OpenCv. Universidad de Crdoba.
16. Smith, R. C., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. The international journal of Robotics Research, 5(4), 56-68.
17. Pupilli, M., & Calway, A. (2005, September). Real-Time Camera Tracking Using a Particle Filter. In BMVC.
18. Szeliski, R. (2006). Image alignment and stitching: A tutorial. Foundations and Trends in Computer Graphics and Vision, 2(1), 1-104.
19. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: an efficient alternative to SIFT or SURF. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2564-2571). IEEE.
20. Morro, A., Sgorbissa, A., & Zaccaria, R. (2011). Path following for unicycle robots with an arbitrary path curvature. Robotics, IEEE Transactions on, 27(5), 1016-1023.
21. Sgorbissa, A., & Zaccaria, R. (2010, October). 3D path following with no bounds on the path curvature through surface intersection. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on (pp. 4029-4035). IEEE.
22. Sgorbissa A., 2015, An integrated approach to path following, obstacle avoidance, and motion planning in 2D and 3D workspaces: a case study with mobile, flying, and underwater robots, DIBRIS, University of Genova, Genova, Italy, Technical Report.
23. Korf, R. E. (1990). Real-time heuristic search. Artificial intelligence, 42(2-3), 189-211.

24. Koenig, S., & Simmons, R. G. (1996, August). Easy and hard testbeds for real-time search algorithms. In AAAI/IAAI, Vol. 1 (pp. 279-285).
25. Baglietto, M., Cannata, G., Capezio, F., Grosso, A., Sgorbissa, A., & 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).
26. Cannata, G., & Sgorbissa, A. (2011). A minimalist algorithm for multirobot continuous coverage. Robotics, IEEE Transactions on, 27(2), 297-312.
27. Balch, T., & Arkin, R. C. (1998). Behavior-based formation control for multirobot teams. Robotics and Automation, IEEE Transactions on, 14(6), 926-939.
28. Recchiuto, C. T., Sgorbissa, A., & Zaccaria, R. (2015, August). Usability evaluation with different viewpoints of a Human-Swarm interface for UAVs control in formation. In Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on (pp. 523-528). IEEE.
29. Chen, J. Y., Haas, E. C., & Barnes, M. J. (2007). Human performance issues and user interface design for teleoperated robots. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37(6), 1231-1245.