Area Inspection Using A Swarm Of UAVS: Preliminary Results.

El Houssein Chouaib Harik[1], *Student Member, IEEE*, François Guérin[2], Frédéric Guinand[3], Jean-François Brethé[2], *Member, IEEE*, Hervé Pelvillain[4], Julien Philippe[5], Julien Kritter[5]

***Abstract*— We present in this paper the preliminary results of an area inspection using a swarm of Unmanned Aerial Vehicles. The map of the inspected area is processed to extract navigable zones. After selecting Points Of Interest (POI) in the image of the inspected area, a seperate navigation path is extracted for each UAV. The UAVs follow the given path using their onboard sensors, and stream in real time a video feedback to the supervision station, where a human operator can at any time assign a new point of interest for the inspection purpose or take back manual control.**

I. INTRODUCTION

The democratization of small Unmanned Aerial Vehicles (UAVs) has allowed a huge leap in mobile robotics. They have been used alone, or coupled with Unmanned Ground Vehicles (UGVs) in a wide range of applications, going from mapping [1], [2], [3], [4], [5], to surveillance and inspection missions [6], [7], [8], [5], [9]. When deployed alone, UAVs may suffer from the limited range of application, particularly when using Vertical Take-Off and Landing (VTOL) vehicles, where their battery life allows in best cases 30 minutes of flying time, which adds the time restriction for a given mission, especially when covering large areas. For this reason, works in [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] have used multiple UAVs to overcome this problem, along side with other challenges that a single UAV may face such as connectivity, coverage, etc... Adding to that, using multiple UAVs is generally the most effective way to successfully complete a task within a reasonable frame of time. Inspired mainly from nature (eg. ants), this cooperation consists in *working together and reasoning about each other's capabilities in order to accomplish a joint task* [20]. Cooperation between UAVs in particular, and in mobile robotics in general, attracts increasingly the attention of research groups. It has opened the field to new applications that could not be possible (or very delicate) to achieve using a single robot system, where the design and consequently the control complexity increases with the mission requirements. In cooperation scheme the robots work together to achieve the predefined goal, and their architecture is kept at simpler level. Nevertheless, cooperation between mobile robots raises different challenges, from task allocation to behavior selection. We present in this paper our first steps towards multiple UAVs cooperation for area inspection mission. The present work covers both preliminary theoretical and experimental aspects towards such an application using multiple UAVs.

We present in section II the autonomous inspection framework using a swarm of UAVs. Followed by the architecture design in section III. The experimental setup will be presented in section IV. The conclusion and future work will be discussed in section V.

[1]EHC. Harik is with LITIS, University of Le Havre, France. E-mail: el-houssein-chouaib.harik@univ-lehavre.fr.

[2]F. Guérin and J-F. Brethé are with GREAH, University of Le Havre, France. E-mail:(francois.guerin,jean-francois.brethe)@univ-lehavre.fr.

[3]F. Guinand is with LITIS, University of Le Havre, France, and UKSW, University Cardinal Stefan Wyszynski in Warsaw. E-mail: frederic.guinand@univ-lehavre.fr.

[4]H. Pelvillain is with IUT GEII, University of Le Havre, France. E-mail: herve.pelvillain@univ-lehavre.fr

[5]J. Philippe and J. Kritter are master students at the University of Le Havre, France. E-mail: (julien.philippe,julien.kritter@etu.univ-lehavre.fr

II. THE INSPECTION FRAMEWORK

To meet real world applications, the presence of a constant supervision of a human operator is primordial. The manual control of any component of the system should be accessible at any time. For this purpose we included a human in our system. The main purpose of the human operator is the high end tasks allocation like the places to inspect, where he has, by a simple click, select the desired Points Of Interest (POI) on the inspected area image (figure 1). We took as an example the site of University of Le Havre (France). The map was taken from a satellite imagery website [21].

Fig. 1. Inspection site

Besides the manual POI selection, another way to perform the inspection mission is the autonomous POI assignment. This scenario is for patrolling the site in a routine basis.

In both cases (manual and autonomous POI selection), the UAVs will be dispatched to these points autonomously. During their flight, the UAVs stream in real time their video feed captured from their onboard cameras, and the human operator will have on the ground station a Closed-circuit television (CCTV) like screen including the streaming of the each UAV assigned with its ID number.

III. Architecture design

The chosen infrastructure is a client/server type. Where a ground station acts like the server that plays the role of the bridge between several UAVs used for the inspection mission, and the supervision devices used by the human operator (figure 2). We developed a java based application that can be run on multiple supervision terminals, which are the devices that can be used by the human operator in order to supervise the fleet and select the patrolling mode (manual or autonomous POI selection).

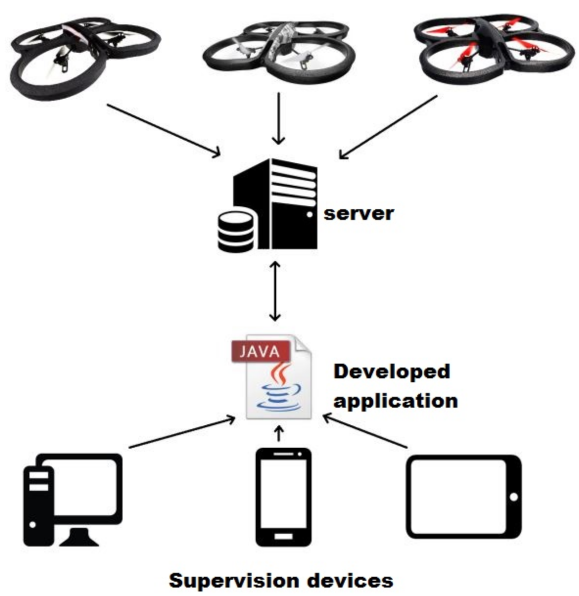

Fig. 2. Global architecture

Each UAV is equipped with a down facing camera to provide a global view of the area. The taken video is streamed in real time to the ground station (server), in order to be fed afterwards to the supervision devices.

A. Autonomous navigation

After selecting POIs on the inspection area image (previously shown in figure 1), the navigation path for each UAV needs to be extracted. A first necessary step consists in extracting the possible path of each UAV. For this purpose different methods are available. Some are texture based [22], others are color based [23], but both presume that the free path are structured roads, which is not the case in hazardous environments. Other methods for ground navigation use monocular vision [24], or stereo-vision [25].

These techniques can be classified according to their usage into two categories, on-line methods, where the information is extracted in real-time to be used for navigation, or off-line methods, where the information is processed prior to the mission in order to be used later. As a first effort towards the autonomous navigation, and for time constraint, we used a simplified, manual off-line method to extract free path spaces.

The image of the area to be inspected is meshed into regular cells. Each cell has at least the size of the used UAV with safety diameter. After that we select manually with the mouse the forbidden zone (black squares in figure 3).

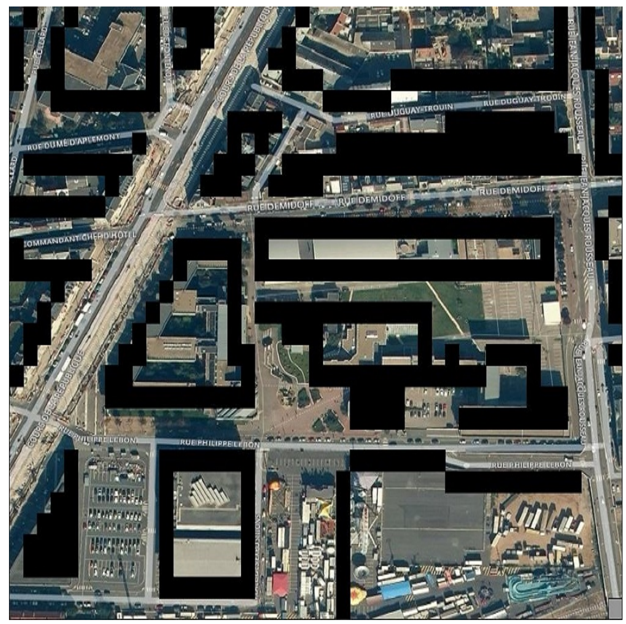

Fig. 3. Free path extraction

1) Path planing: After selecting the obstacles in the image of the inspected area, the path for the UGV is then extracted using a shortest path algorithm based on its actual location and its destination. Shortest path algorithms are a popular subject within computer science community, and the literature is rich with those algorithm. Two of the most popular shortest path algorithms are A star (A*) [26] and Dijkstra [27].

Consider an area to be inspected regularly, the best solution to implement in this case is the Dijkstra algorithm. Considering one cost function to move from a cell to another, Dijkstra's algorithm calculates the shortest path from the source, supposed to be the control tower where the human operator is located, and where the vehicles are located initially, to each cell in the zone. This will avoid to process the map again unless the human operator wants to inspect another area, in this case the source will be the actual location of the UGV.

We implemented in Processing [28] a Dijkstra like algorithm. After extracting the free path zones, the human operator selects the destination of the UAVs, and the shortest path to that location will be extracted (figure 4).

Fig. 4. Trajectory generation

The extracted shortest path will be converted into navigation waypoints to be used by the UAVs. These waypoints can be represented by metric values if the body reference of the UAVs is used (odometery navigation), or GPS coordinates if the signal is available. Please note that another restriction should be taken in consideration when performing this algorithm for multiple UAVs, where to avoid collisions, after extracting the path of each UAV, the corresponding cells are turned into occupied cells, and cannot be further used for the next UAV shortest path navigation.

IV. Experimental results

For the experiments we have used an off-the-shelf commercially available UAV, which is the AR Drone 2.0 from parrot [29]. This choice of using the AR Drone is based on its cost, simplicity, and the available Software Development Kit (SDk) provided by the manufacturer, as well as the large community working with this platform, which provides a solid basis for troubleshooting.

A. Communication protocols

The communication with the AR Drone goes through different canals both on Transmission Control Protocol (TCP), or User Datagram Protocol (UDP). This is a bidirectional communication.

Once the UAVs connected to the server, the following services are available:

- The UAV configuration (UDP, port 5556). The usage of ATtention commands (AT commands) allow the modification of the configuration of the UAV for the calibration and piloting purposes.
- Navigation data (*navdata*, UDP, port 5554). Once the UAV is online, the navigation data (IMU feedback, speed, altitude, etc.) can be received through this port.
- Video flow (TCP, port 5555). The AR Drone 2.0 is equipped with an HD (720p) front facing camera, and a VGA down facing camera. The flow of these cameras can be received on this port separately (one camera flow at a time).
- Reading UAV configuration (TCP, port 5559). The UAV sends its configuration file containing its different parameters through this port.

B. Topology

The default topology of the AR drone 2.0 is in mode *master*, in which the UAV acts as an access point, and the client connects to this access point in order to communicate with the AR Drone through the previously mentioned protocols and ports (figure 5.

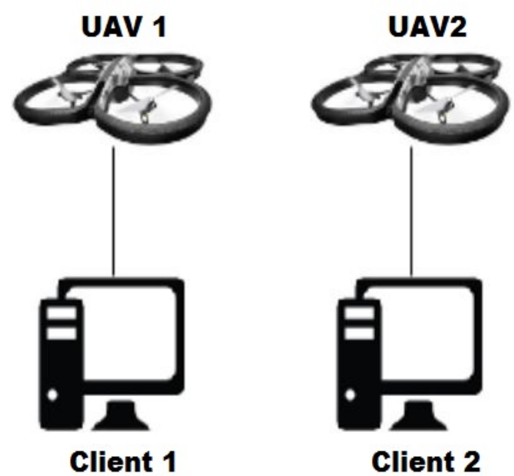

Fig. 5. Single user topology (default)

We can notice that each client (each network interface) connects to a separate UAV. It is clear that this mode cannot be used for our application, where several UAVs are present in the architecture, and following the default configuration, the client should have several network interfaces in order to connect to each UAV separately.

To overcome this problem, we used the infrastructure topology (figure 6). The UAVs as well as the supervision devices connect to the access point, which will be in charge of the routing operations of the network. This model corresponds to the desired inspection application, where a human operator can supervise several UAVs at the same time.

The clients in figure 6 are the supervision devices running the developed Java application, which can be any Java enabled running terminal, such as computers, smartphones, tablets, etc...

Despite the indications given by the documentation provided by Parrot, ad-hoc modes and infrastructure are not implemented on the drones themselves. Whatever the configuration done via AT commands, or via telnet, the drone will always act as an access points. A bug in the implementation, makes the UAV completely inaccessible if the mode is set to 1 (ad-hoc) or 2 (infrastructure).

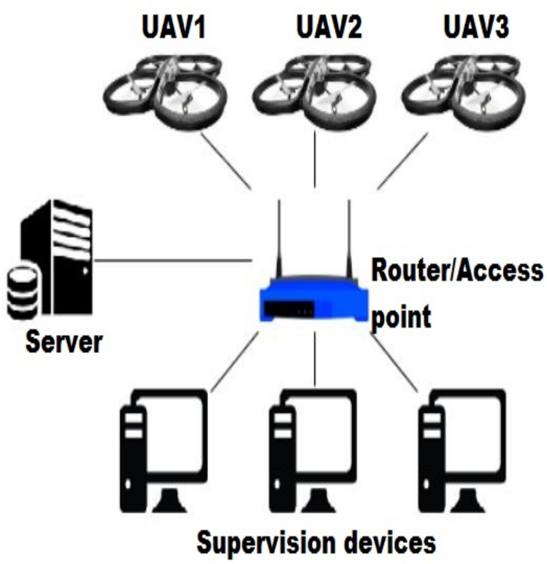

Fig. 6. infrastructure topology

The network configuration is performed via **/bin/wifi_setup.sh** script, which is launched when booting via **/etc/inittab** and **/etc/init.d/rcS**. A need step consisted in rewriting and adapting these scripts to complete the implementation of the infrastructure mode (2), allowing the UAVs to join an access point instead of acting like one themselves.

During the test phase, a *watch dog* script was included via **/etc/init.d/rcS** in order to maintain access to the UAV if a network failure is present. In case of a problem with the router, the *watch dog* role is to reset the AR Drones' configuration after 5 minutes of inactivity, and this to gain access again to the UAV.

Once the server is launched, and the UAVs as well as the supervision device are connected to the access point, and ID is assigned to each UAV, and the client (supervision device) has the complete control of the UAVs whether individually, or as a group (swarm).

At this stage, the human operator can send the required waypoints for the inspection task, or we can run a script that assign these waypoints autonomously using the previously presented techniques for path planning.

The initial experiments included three AR Drone 2.0 UAVs, a LinkSys router, and two laptops (one acts as the server, and the second as the supervision device). We were able to perform the basic moves (take-off, landing, pitch, roll, yaw, and altitude) from the client interface controlling at the same time the three UAVs (figure 7).

Fig. 7. UAVs testbed

V. Conclusion and Perspectives

We presented in this paper the preliminary results towards autonomous area inspection using swarm of UAVS. A human operator, through the supervision device, can control the swarm by assigning individual waypoints, or selecting the patrol mode for autonomous waypoints assignment. The navigation areas' image is processed to extract the free path navigation. After the waypoints selection (manually or autonomously), a shortest path algorithm is applied, and then sent to the selected UAV as metric of GPS coordinates. To control the swarm of UAVs we developed a server/client application. The used UAVs acts by default as access points, thus the UAVs internal network interfaces has been modified in order to allow the implementation of the infrastructure topology, allowing several UAVs to connect to the same access point. The first experiments consisted in performing the basic movements of three UAVs (take-off, landing, pitch, roll, yaw, and altitude) and this from a single supervision device.

The perspectives of the present work are two folds. Firstly, we would like to include a time variant in the path planning. This will allow the intersection of the UAVs paths without collision, which will reduce the flying time of the UAVs by reducing their traveled paths. The second perspective of the work consists in experimenting the path planing outputs with the UAVs in an outdoor environment to check the efficiency of the proposed architecture for an inspection mission.

VI. Acknowledgments

The authors would like to thank Le Havre Town Council CODAH for their support under research grants. They would like as well to thank Haute-Normandie Region (France) through their support to PCMAI project.

References

[1] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, and Satochi

Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.

[2] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3971–3978. IEEE, 2013.

[3] H Jin Kim, Rene Vidal, David H Shim, Omid Shakernia, and Shankar Sastry. A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 634–639. IEEE, 2001.

[4] René Vidal, Shahid Rashid, Cory Sharp, Omid Shakernia, Jin Kim, and Shankar Sastry. Pursuit-evasion games with unmanned ground and aerial vehicles. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2948–2955. IEEE, 2001.

[5] M Ani Hsieh, Anthony Cowley, James F Keller, Luiz Chaimowicz, Ben Grocholsky, Vijay Kumar, Camillo J Taylor, Yoichiro Endo, Ronald C Arkin, Boyoon Jung, et al. Adaptive teams of autonomous aerial and ground robots for situational awareness. *Journal of Field Robotics*, 24(11-12):991–1014, 2007.

[6] Matthew Ridley, Eric Nettleton, Salah Sukkarieh, and Hugh Durrant-Whyte. Tracking in decentralised air-ground sensing networks. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 1, pages 616–623. IEEE, 2002.

[7] Ben Grocholsky, Rahul Swaminathan, James Keller, Vijay Kumar, and George Pappas. Information driven coordinated air-ground proactive sensing. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2211–2216. IEEE, 2005.

[8] Cai Luo, Andre Possani Espinosa, Danu Pranantha, and Alessandro De Gloria. Multi-robot search and rescue team. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 296–301. IEEE, 2011.

[9] Herbert G Tanner and DK Christodoulakis. Decentralized cooperative control of heterogeneous vehicle groups. *Robotics and autonomous systems*, 55(11):811–823, 2007.

[10] Martin Rosalie, Grégoire Danoy, Serge Chaumette, and Pascal Bouvry. From random process to chaotic behavior in swarms of uavs. In *6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, 2016.

[11] Pascal Bouvry, Serge Chaumette, Grégoire Danoy, Gilles Guerrini, Gilles Jurquet, Achim Kuwertz, Wilmuth Müller, Martin Rosalie, and Jennifer Sander. Using heterogeneous multilevel swarms of uavs and high-level data fusion to support situation management in surveillance scenarios. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, 2016.

[12] Martin Rosalie, Gregoire Danoy, Pascal Bouvry, and Serge Chaumette. Uav multilevel swarms for situation management. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 49–52. ACM, 2016.

[13] Hu Zhi-wei, Liang Jia-hong, Chen Ling, and Wu Bing. A hierarchical architecture for formation control of multi-uav. *Procedia Engineering*, 29:3846–3851, 2012.

[14] Rachael Purta, Mikolaj Dobski, Artur Jaworski, and G Madey. A testbed for investigating the uav swarm command and control problem using dddas. *Procedia Computer Science*, 18:2018–2027, 2013.

[15] Sebastian Rohde, Markus Putzke, and Christian Wietfeld. Ad hoc self-healing of ofdma networks using uav-based relays. *Ad Hoc Networks*, 11(7):1893–1906, 2013.

[16] Yi Wei, M Brian Blake, and Gregory R Madey. An operation-time simulation framework for uav swarm configuration and mission planning. *Procedia Computer Science*, 18:1949–1958, 2013.

[17] Tae Soo No, Youdan Kim, Min-Jea Tahk, and Gyeong-Eon Jeon. Cascade-type guidance law design for multiple-uav formation keeping. *Aerospace Science and Technology*, 15(6):431–439, 2011.

[18] Liguo Weng, Qingshan Liu, Min Xia, and YD Song. Immune network-based swarm intelligence and its application to unmanned aerial vehicle (uav) swarm coordination. *Neurocomputing*, 125:134–141, 2014.

[19] Arpita Sinha, Antonios Tsourdos, and Brian White. Multi uav coordination for tracking the dispersion of a contaminant cloud in an urban region. *European Journal of Control*, 15(3):441–448, 2009.

[20] Lynne E Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1):5–14, 2008.

[21] Coordinate systems worldwide, 2016. URL: http://epsg.io/.

[22] Jinyou Zhang and H-H Nagel. Texture-based segmentation of road images. In *Intelligent Vehicles' 94 Symposium, Proceedings of the*, pages 260–265. IEEE, 1994.

[23] Jill D Crisman and Charles E Thorpe. Unscarf-a color vision system for the detection of unstructured roads. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2496–2501. IEEE, 1991.

[24] S Wybo, D Tsishkou, C Vestri, F Abad, S Bougnoux, and R Bendahan. Monocular vision obstacles detection for autonomous navigation. In *IROS*, page 4190, 2008.

[25] Kurt Konolige, Motilal Agrawal, Robert C Bolles, Cregg Cowan, Martin Fischler, and Brian Gerkey. Outdoor mapping and navigation using stereo vision. In *Experimental Robotics*, pages 179–190. Springer, 2008.

[26] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[27] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[28] Processing, 2016. URL: https://processing.org/.

[29] Parrot. Quadcopter ar drone 2.0, 2016. URL: http://ardrone2.parrot.com/.