Galileo Aid Drone: A System Integration for Autonomous Wildfire Assistants

Diego Calderón, Rodrigo Cordero, Ariel González, Ali Lemus and Julio Fajardo

Abstract— **Mobile robots are playing an important role as monitoring and immediate response services supporting risk management systems for natural or artificial disasters. In this work, an integration of systems based on a Unmanned Aerial Vehicle collects relevant information and show the status of a wildfire through an interactive dashboard, which can be accessed by the risk and disaster management systems through any computer. In addition, the system can identify people and communities at risk by presenting their estimated geolocation on a map, as well as the spread of the fire. In this way, it will be possible to understand the magnitude of the disaster, and thus develop better rescue and monitoring strategies.**

Monitoring system, risk management, rescue robots.

I. Introduction

Wildfires are a serious problem that has both economic and environmental effects to the world, putting in danger people, animals and property that are near or inside of the affected area. They produce an economical lost leaving people without their homes as well as environmental damage, such as the water contamination in the near rivers [1]. The first responders struggle with several issues in order to control the fire and to evacuate the zone to prevent the loss of human lives and property. Some of the more challenging issues is to monitor and predict the spread of the fire in order to establish a plan to find and rescue the humans near the affected area. Autonomous robots have become more popular in recent years to assist first responders in natural and artificial disasters, specially the Unmanned Aerial Vehicles (UAVs). This, because of its small size, relatively low-cost components and due its ability to carry cameras and sensors on board that can help firefighters to explore the fire-affected area from an aerial view. In this way, the UAV can acquire relevant information about the wildfire and start looking for hot spots such as the geolocation of people and communities on risk, identify the best routes to evacuate them and plan better strategies to contain the fire based on the observations acquired by the system.

In this work, a quadrotor-based solution is presented to assist first responders during a wildfire, giving them the ability to quickly and continuously explore and monitor the affected area using an interactive dashboard, which can be accessed by the risk and disaster management systems using any computer with a Linux distribution.

II. System Architecture

The architecture of this solution, as shown in Fig. 1, is based-on a quadrotor with a Fligh Managment Unit (FMU) RDDRONE-FMUK66, which has as NXP Kinetis K66 at 180MHz Microcontroller Unit (MCU), that runs the PX4 open source autopilot software stack [2]. Also, there is a NXP NavQ as a companion computer which has the following specifications: 2GB of RAM, a Quad-Core ARM A53 at 1.8 GHz which runs a lite version of Ubuntu 20.04 Operating System. On other hand, there are three cameras in the system, the first one is an Intel RealSense D435i depth camera, which provides RGB-D images (depth sensor) and orientation information through its integrated inertial measurement unit (IMU); this camera is connected to the NavQ using a USB type C port. The second one, is a FLIR Lepton 2.5 Longwave Infrared (LWIR) Micro Thermal Camera, which is connected to the NavQ board using its SPI and I2C ports that are available on the NavQ board and finally, there is a Google Coral RGB Camera that is also connected to the NavQ using the MIPI CSI interface.

Furthermore, the system integration is made using the Robotic Operating System (ROS) [3]. Thus, the rosmaster and nodes to communicate with the cameras and the FMU are running in the NavQ single board computer. In order to detect people, animals and property at risk, YOLOv3 was employed due its speed and lightweight, making it a good option to be executed in a single board computer [4].

In a similar way, different computers can run ROS in a distributed way, in order to connect to the ROS master that is running in the NavQ and to access the information that is published by the different nodes running on the NavQ. Additionally, Mapviz is used as a visualization tool to show a dashboard with the information that the drone is gathering during the flight.

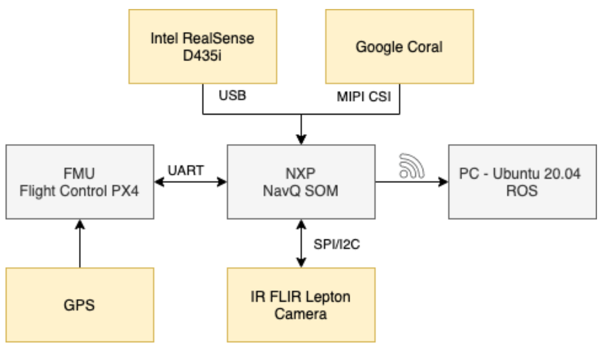

Fig. 1. Block diagram of the system architecture implemented on the drone.

A. Hardware

1) Quadrotor: The UAV (shown in Fig. 2) is based on a carbon fiber mechanical frame of approximately 500 mm

Fig. 2. Galileo Aid Drone during a test flight.

of diagonal size with four Brushless DC (BLDC) motors of 920 kV with its respective Electronic Speed Controlles (ESCs). The FMU is based on an ARM Cortex-M4 MCU and is supported by the business-friendly open-source PX4 flight stack. This software is in charge to manage the BLDC motors drivers and the GPS NEO-M8N module. On the other hand, the system has two RF communications modules, a telemetry radio HGD-TELEM915, to establish communication with any computer or tablet via MAVLink protocol and another to connect with the remote control. Moreover a 4200mAh Lithium-Polymer Battery supplies power to the system.

2) NXP NavQ: A companion computer is added to leave the FMU just in charge of flight management. This single board computer was chosen because it is designed to be mounted on mobile robots having multiple useful peripheras, providing access to I2C, UART and SPI communication protocols which are more convenient to communicate with the FMU and cameras. Furthermore, it's capable of running ROS nodes such as *mavros* to communicate with PX4 stack using a UART port, and it provides WiFi and Bluetooth connections that are useful to share information with other computers.

3) Cameras: The system has two mounted cameras to acquire aerial views of the scene, the first one is the LWIR FLIR Lepton 2.5 camera to obtain and monitor thermal images of the fire with a resolution of 80x60 pixels and a spectral range with a longwave infrared of 8 μm to 14 μm see Fig. 3(b), and the second one is the Google Coral RGB camera that is used to detect people and property at potential risk. In addition, to avoid possible collisions with obstacles, an Intel RealSense D435i is mounted. This camera provides RGB and Depth images with a resolution up to 1920x1080 pixels at 30 fps for RGB and 1280x720 pixels at 90 fps for Depth.

B. Software

1) YOLOv3: This module is used because it is a state-of-the art solution to do real-time object detection in images using limited resources and it still very fast. Specifically, a custom CNN was trained based on a subset of the COCO Dataset using the categories that are more important to identify in a disaster such as people, animals and property as shown in Fig. 3(a).

2) Objects/People Detection and Localization: One of the key tasks of the system is keeping track of the people/property. This task is implemented using the information from the bounding boxes that are provided by YOLO as well as feature matching techniques for aerial images. Moreover, the characteristics of the camera and the data from the GPS (longitude, latitude and altitude) are considered to make an estimation of the coordinates of an object or a person and its added to the system to keep track of it.

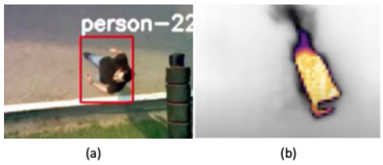

Fig. 3. Galileo aid drone aerial view. (a) Identification of a person using YOLOv3, (b) Thermal image of an small fire.

3) Dashboard: An interactive interface that displays the collected information during the flight is implemented to support risk management systems. Specifically, it shows the geolocation of the UAV into a map using the Global Positioning System (GPS) information. The trajectory of the UAV is drawn in the map using a red line and markers show the location of people and animals. The video that is being captured from the Google Coral camera that is mounted on the drone is displayed in the top left corner as shown in Fig. 4.

Fig. 4. Dashboard based on MapViz to see the markers and the drone position

III. Conclusion

The system integration proposed in this work can successfully support first responders in disaster situation. The utilization of a distributed system like ROS facilitates the management and processing of the information displayed in the dashboard. Moreover, since it is a distributed system it can be scaled to use multiple UAVs and integrate the information in a single dashboard.

References

[1] C. Santín, S. H. Doerr, X. L. Otero, and C. J. Chafer, "Quantity, composition and water contamination potential of ash produced under different wildfire severities," *Environmental Research*, vol. 142, 2015.

[2] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6235–6240.

[3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009.

[4] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018.