2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Low-cost quadrotor applied for visual detection of landmine-like objects

J. Rodriguez[1], C. Castiblanco[2], I. Mondragon[3] and J. Colorado[4]*

Abstract— **This paper presents the use of a low-cost quadrotor applied for visual detection of landmine-like objects. Nowadays, landmines in Colombia are harmful and even letal artifacts mostly abandoned in rural areas. A percentage of the overall number of landmines are hand-crafted and partially exposed on the terrain's surface so that they can be triggered. Based on that fact, we propose an artificial vision approach as a complementary tool for landmine detection. To this purpose, we have developed an open-source software package based on the Robot Operating System (ROS) platform that can be easily: (i) integrated with a low-cost quadrotor such as the AR.drone parrot 2.0 and (ii) applied for real-time visual detection of landmine-like objects (tuna cans). This article describes the ROS-based package called *drone_detection* and the experiments carried out for assessing the detection performance for different type of terrains and landmine visibility.**

I. INTRODUCTION

The procedure of explosive landmine detection and removal is referred to as demining. Several tools have commonly included metal detectors [1]–[3], trained animals or machines that rely on human intervention [4], [5]. Only in 2011 were reported 4.286 victims wordwide, whereas in Colombia, authorities have reported 10.253 victims since 1990 [6]. Within the Colombian scenario, a large percentage of these artifacts is mostly abandoned in rural areas and the military is carrying out an exhausting labour for demining larger areas and keeping civilians safe [7]–[10]. They typically use terrestrial platforms that are expensive, require high-qualified personnel, slow for covering larger areas and even dangerous in cases where the detection process is compromised [11]–[14].

This paper presents a step towards the use of small-size low-cost UAV (Unnamed Aerial Vehicles) as a promising complementary tool for aerial landmine detection [15]–[18]. Our goal is to evaluate whether visual recognition methods are reliable for aerial detection of landmine-like objects particularly within the Colombian context. Since most of these landmines are hand-crafted using materials such as tuna cans or plastic bottles, they have to be partially exposed for being triggered. In this paper, we introduce an open-source package (*drone_detection*) based on the Robot Operating System (ROS) [19]. The package[1] contains vision algorithms that process sequences of images captured by the bottom camera onboard the UAV (cf. Fig 1b). We have used the commercial quadrotor AR.drone 2.0 [20], a low-cost UAV platform equipped with two cameras onboard, data-link and Inertial Measurement Unit (IMU) sensor.

This article is divided in three main sections. Section II briefly details the Guidance-Navigation-Control (GNC) station which enables the mission control. Section III introduces the methods used for landmine detection and how the *drone_detection* ROS-based package can be easily integrated within the GNC. Section IV describes the experimental scenarios and presents how the proposed system is able to operate in difficult terrains with rocky and large grass, dry leafs and partially buried tuna-cans acting as landmine-like objects.

II. THE INTEGRATED DETECTION TOOL

A. Parrot AR.Drone 2.0

The Parrot AR.Drone commercial quadrotor [20] is a low-cost platform with a light structure (mass of $420g$) and a robust mainframe able to resist unexpected crashes. The drone is equipped with two cameras onboard (one aims to the ground), an IMU with 3-axis gyroscope/accelerometer and an ultrasound sensor for measuring altitude. The camera that aims downward has a field of view of 47.5^o with a resolution of 320×240 pixels at 60fps.

In previous work [21], we used the AR.Drone as a testbed for capturing images of tuna-can objects acting as landmines. An initial approach for visual detection was introduced and assessed by varying the flight speed and the altitude of the drone. In this work, we have integrated the AR.Drone with a GNC station that automatically manages the visual detection task.

B. GNC Station

It is basically composed by a laptop running the Robot Operating System (ROS), which consists of a flexible framework that provides tools and libraries for writing robot software. Figure 2 shows the architecture of the GNC station. The modules are ROS-packages coded in C++ programming language, as described in the following:

- *Connection:* handles the communication between the GNC and the drone over the Wi-Fi network by means of a specific driver developed by Monajjemi et al., [22]. The driver called *ardrone_autonomy* is an open-source ROS node that enables the access to the AR.drone

This work was supported by the SIRP and CEPIT research groups of the School of Engineering at Pontificia Universidad Javeriana (PUJ) in Bogota, Colombia. First and second authors have equally contributed in this research.

[1]is research assistant at the SIRP group in PUJ, j_rodriguez@javeriana.edu.co

[2]is research assistant at the SIRP group in PUJ, jenny.castiblanco@javeriana.edu.co

[3]is an assistant professor at the Dept of Industrial Engineering at PUJ, imondragon@javeriana.edu.co

[4]*Corresponding Author. He is an assistant professor at the Dept of Electronic Engineering at PUJ, coloradoj@javeriana.edu.co

[1]Available at https://code.google.com/p/uav-detection/

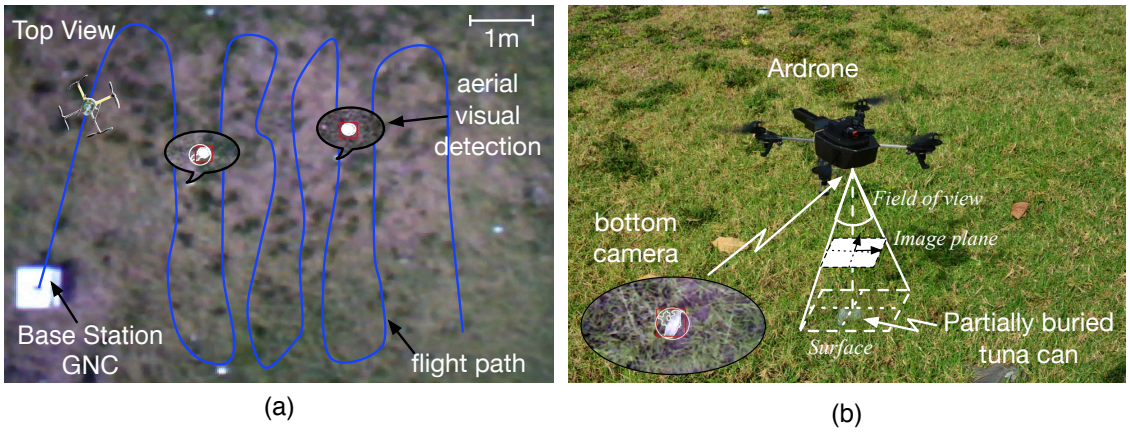


Fig. 1. (a) Aerial top view of one terrain covered by the AR.drone, (b) description of the landmine-like object visual detection.

sensor data from the GNC station wirelessly (cameras and IMU).

- *Navigation:* contains the drone autopilot (high-level attitude controller). It enables the operator to remotely teleoperate the drone from the GNC station. This module's features are provided by the open-source ROS node *Tum_ardrone* [23].
- *Detection:* uses the OpenCV library [24] for image recognition, noise filtering, landmine detection, recording live video of the terrain and feedback drone data to the user in real time (cf. Fig 5). These features are provided by the *drone_detection* ROS package developed in this research.

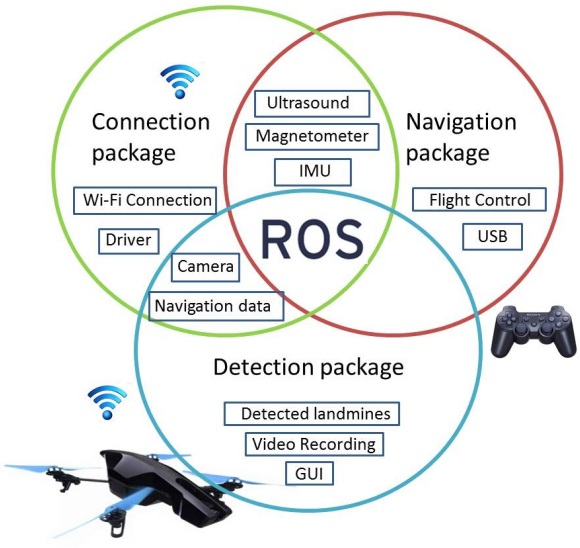


Fig. 2. Architecture for visual landmine-like object detection using the AR.drone.

III. DRONE DETECTION PACKAGE

Figure 3 shows a flowchart describing the modules that compose the proposed *drone_detection* package: navigation, control, detection and record.

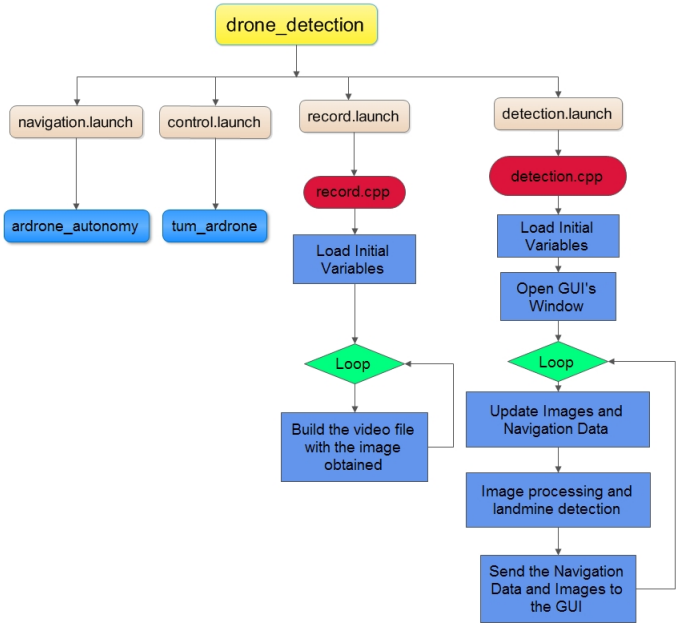


Fig. 3. Flowchart of the proposed *drone_detection* ROS-package. This package depends on the open-source packages described in [25] and [22].

The navigation module allows for reading the bottom camera of the AR.drone. As shown in Fig. 1b, the detected object (tuna can) is projected onto the image-plane frame of reference. The *ardrone_autonomy* node [22] publishes a resultant image that has been transformed after estimating camera rotation and translation. The resulting image is received from one video stream encoded with 8-bits YCBCR color space and using the H.263 UVLC codec with a bandwidth of $10kB$ per frame. The received image is then processed by the detection module that identifies whether the object is a landmine or not.

The *Tum_ardrone* ROS node contains the control module. Autonomous flight is based on: monocular SLAM [26], extended Kalman Filter (EKF) and Proportional-Integral-Derivative (PID) control strategy. The monocular SLAM technique has not been used in this research, however, it re-

ceives video frame from the front camera of the AR.drone at $18Hz$. This could be useful for autonomous indoor navigation and mapping. The EKF technique is useful for drone's pose estimation based on sensor measurements (IMU, ultrasound) at $200Hz$ and feedback the current state of the AR.drone (attitude and position) to the PID controller. The PID provides control commands at $100Hz$.

The *drone_detection* package is composed by two main algorithms: detection and record. The former contains the visual approach for post-processing the acquired image and performing object recognition, whereas the latter saves images, video and flight data during the mission. The system also generates a log data file containing key information of the flight.

A. Detection algorithm

The object recognition task is based on classical vision algorithms [27]–[29] such as noise filtering, image binarization and morphologic operations. The functions that support these features are available at [30] and they have been coded within three main C++ functions named: (i) Navdata_Uload(), (ii) Image_Upload() and (iii) detection_landmines().

- *Navdata_Uload() and Image_Upload():* both are threads that receive the flight data from the drone. The former deals with the IMU readings, estimated velocity and battery level, whereas the latter loads the latest image received from the drone's bottom camera.
- *detection_landmines():* is the core function of the detection module (cf. Fig 3). It carries out the detection task via three main procedures applied to the acquired image: (i) noise filtering, (ii) feature extraction and (iii) classification process. This last one determines whether the detected object in the image corresponds to a landmine or not.

Noise filtering

The noise filtering function *Image_Filtering()* removes from the image all the objects that do not match the landmine characteristics. Firstly, the original image is transformed to a grayscale space for performing binarization and morphological operations. Secondly, both operations are used for removing small objects from the image (noise) by means of applying erosion and dilation processes, as shown in (1) and (2) respectively.

$$A \ominus B = \bigcap_{b \varepsilon B} A_{-b} \tag{1}$$

$$A \oplus B = \bigcup_{b \varepsilon B} A_{b} \tag{2}$$

Erosion or dilation is applied to the image A, while B is the morphological element that proceeds. Erosion removes pixels on the object boundaries, while dilation adds pixels to the boundaries of objects in an image. Once the erosion operation is applied, we use the function *Remove_Medium_Objects()* to remove medium objects from the image, i.e, the objects that are smaller than a pre-defined pixel-size threshold.

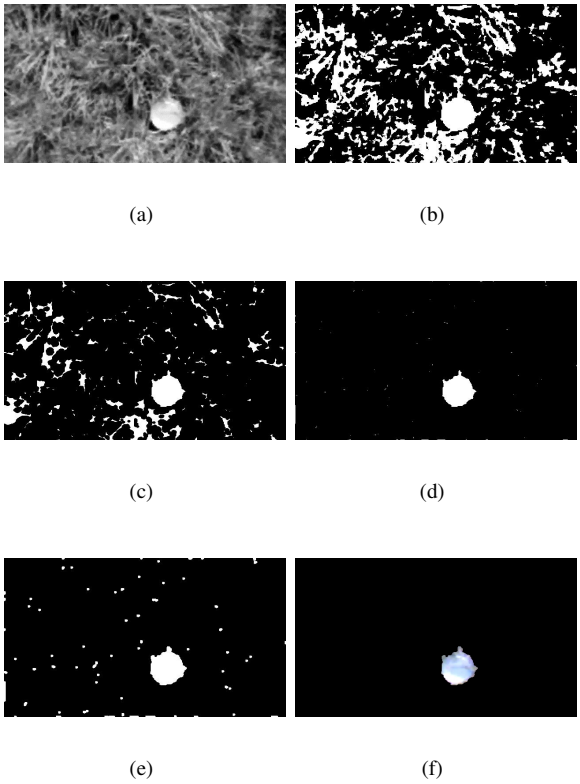


Fig. 4. Transformation process: (a) RGB space to grayscale and (b) grayscale to binary space. Noise filtering process: (c) image after the use of the *erode()* function, (d) image after the removal of medium objects, (e) image after the use of the *dilate()* function, (f) output-image after applying the contours operation.

Figure 4a-b depicts the transformation of the image to grayscale and binary space. Figure 4c-f details the noise filtering process: plot c is the resultant image after applying the *erode()* function for removing small objects, plot d is the resultant image after applying the *Remove_Medium_Objects()* function for removing medium objects, plot e depicts how the *dilate()* function allows for recovering those removed fragments of the landmine after the erode process, whereas the plot f is the resultant image with an isolated detected object.

Feature extraction

The feature extraction function *Image_Features()* is used to determine whether the output-image from Fig. 4f matches with the characteristics of a landmine-like object or not. It uses the OpenCV *matchtemplate()* function to compare the output-image against a template-image of a known landmine object. A set of indicators that result from that comparison is used as the first feature. The second feature results of using the *Biggest_Area()* function, which computes and compares the area of the output-image. The third feature corresponds to the number of pixels contained within a defined percentage of intensity in each layer of the RGB scale of the image. The percentage of intensity was determinate by analyzing the pixels in each RGB layer using different templates of

Fig. 5. GUI components: (upper-left) image received from the bottom camera of the AR.drone, (lower-left) navigation data and output-image with the detected landmine-like object, (lower-right) live video received from the AR.drone.

landmines as testing elements.

Classification process

The *Classification_Process()* function enables the detection algorithm to determine whether the detected and processed object finally corresponds to the target or not. It uses combinatory logic conditions based on error measurements calculated with the features extracted from the image and the original features of the template. Figure 5 shows the resulting image after applying the noise filtering and the feature extraction functions.

B. Record

It is a simple algorithm that enables the GNC station to record video using the stills obtained by the drone. The images are acquired by subscribing the record program with the camera ROS-topic generated by the *ardrone_autonomy* driver. Video is visualized by the user in real-time using the GUI of the GNC station, as shown in the lower-right inset of Fig 5.

IV. RESULTS

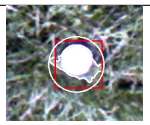
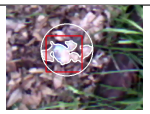
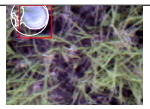
Outdoor experiments have been carried out to analyze whether the proposed vision approach could be used as a complementary tool within a real humanitarian demining task. To this purpose, two kind of experiments were conducted to evaluate the system performance:

1) Percentage of detection in different type of terrains.
2) Percentage of detection of partially buried landmine-like objects.

A. Experimental Scenarios

Most of the experiments were carried out in rural locations nearby Bogota, Colombia. Figure 1a shows the top view of one of the terrains covered, corresponding to an average area of approximately $100m^2$. More difficult terrains with larger grass, rocks, and dry leafs were also located in rural areas similar to those found in the regions of conflict within Colombia. The main setup for the experiments is described as follows:

TABLE I
TYPE OF TERRAINS COVERED BY THE AR.DRONE

Type	Terrain	Description
1		Short green grass
2		Large green grass
3		Large green grass with dry leafs
4		Dry grass
5		Rocky and dry terrain

- Five terrain topologies were selected for the outdoor experiments, as shown in Table I.
- Tuna cans acting as landmine-like objects were placed along each type of terrain.
- The data reported in this paper was acquired during 30 days of experiments. Uncertainties such as sunlight changes and wind speed variations were taken into account.
- The average drone flight speed was $0.3m/s$ with an average flying altitude of $1m$. Both values were determined in previous work [21].
- Images are captured by the AR.drone at a rate of $15Hz$ and processed in the GNC station in real-time ($250ms$). The entire detection process is carried out automatically.

The percentage of detection is measured by obtaining the Receiver Operating Characteristic (ROC) curve, which it is a basic classifier that defines the fraction of true positives out of the overall real positives (TPR = true positive rate) and the fraction of false positives out of the overall real negatives (FPR = false positive rate), as shown in Eq. (3),

$$TPR = \frac{TP}{TP+FN}, FPR = \frac{TN}{FP+TN} \quad (3)$$

where (TP) is the number of true positives, (FN) false negatives, (TN) true negatives and (FP) false positives. The ROC curves obtained for the forthcoming experiments were calculated using Eq. (3) with an overall of 7683 sampled images.

B. Experiment 1: different type of terrains

Figure 6 shows a collection of stills captured during the experiment. The results reported in Fig. 7 show that the terrains with the higher percentage of detection ($> 88.24\%$) correspond to the types: 1, 2 and 5 (cf. Table I). For instance,

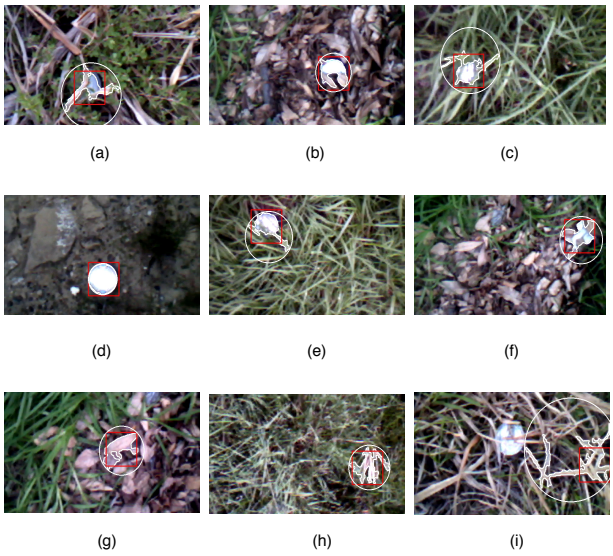


Fig. 6. Stills captured by the AR.drone during experiments (1) and (2). Plots $(a)-(f)$ depict how the proposed system is able to detect partially buried tuna cans distributed in different type of terrains (True Positives). Plots $(g)-(i)$ show typical failures of the system corresponding to (FP) and (FN) respectively.

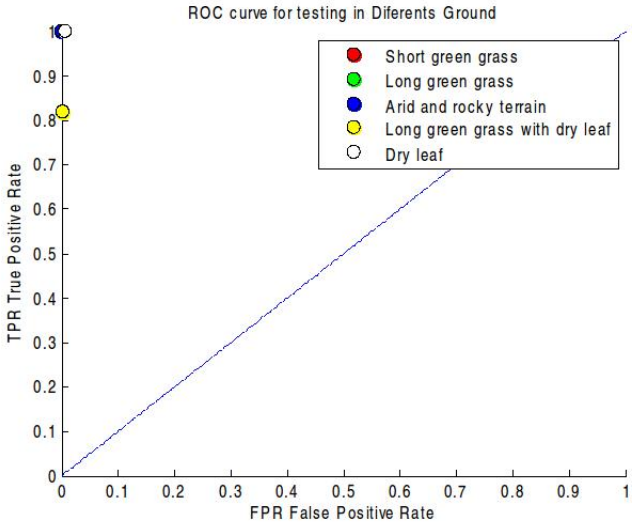


Fig. 7. ROC curve for different terrain topologies.

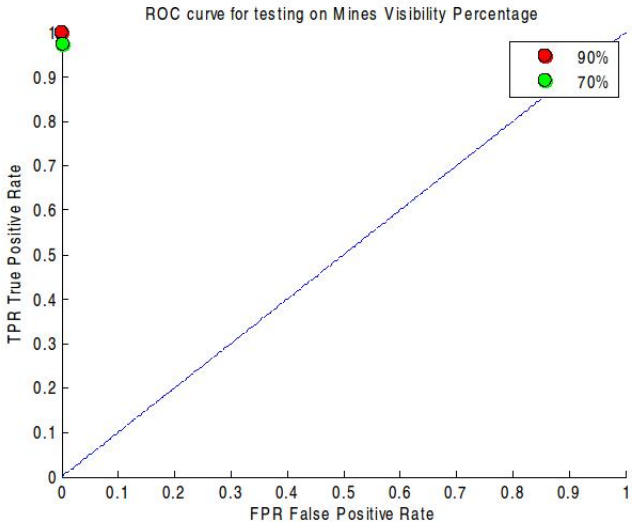


Fig. 8. ROC curve for objects with 90% and 70% of visibility.

dry terrains with leafs have turned out to be difficult scenarios for performing an accurate visual detection. Based on the numerical data, the percentage of detection corresponding to the terrain types 3 and 4 is around 73%. The reason to that is related to the brightness of the rocks and the dry leafs, which are mostly within the brightness range of the image template. Numerically, the minimum intensity percentage that must be satisfied for each pixel of the image is 69.5% in green scale, 75% in blue scale and 75.5% in red scale. These values were determined experimentally.

C. Experiment 2: partially buried landmine-like objects

In this test, the landmine-like objects were placed: *visible*, which corresponds to a range of visibility between 71% and 90% of the object on the surface, and *partially visible*, which corresponds to the range between 30% and 70%. Therefore, the minimum threshold-visibility allowed by the detection algorithm is about 30%. Figure 6a shows a *partially visible* object with 30% of visibility.

Figure 8 shows the corresponding ROC curve for this test. On average, the detection algorithm enables a percentage of detection of 87% for *visible* objects. However, when the object visibility is less than 70%, the number of false negatives (FN) increases, i.e, the detected object is classified as a potential landmine, but at the end, it turns out to be another object.

V. CONCLUSIONS

Based on the results reported in both Fig 7 and 8 one can note that the numerical values are concentrated at the top left-hand border of the ROC space, which demonstrates the accuracy of the tests. Accuracy is measured by the area under the ROC curve: an area of 1 represents a perfect test, whereas 0.5 represents a worthless test. On average, the percentage of detection for partially buried landmines-like objects at any of the terrains (70% of visibility) is about 80%. One key aspect for achieving this result is due to the feature extraction process that classifies the intensity of the object in each RGB layer of the image.

The AR.drone is an affordable platform that can be easily used by everyone. It offers many benefits in terms of usability but constrains the development of more sophisticated visual algorithms due to the lack of computing resources. Despite the detection process still has some issues with objects that have similar size and intensity compared to the features of the template, our system is able to operate in complicated scenarios with similar characteristics to those found in a real demining problem in Colombia.

In order to improve the detection for different land-

mine morphologies, we need to analyze how the change in brightness can be identified within the algorithm and how to compensate these changes during the classification phase. Also, new object's templates with different shapes and materials should be added to the database for enhancing the feature extraction phase of the algorithm. We encourage researchers to download the ROS package available at [30] and to provide feedback that enhances the visual detection methods without compromising the real-time response of the system.

Current work is oriented towards the integration of image mosaicing techniques for mapping the entire covered terrain. Also, geo-localization methods based on GPS data will allow for referencing the position of each detected landmine object within the assembled map. Furthermore, more precisely sensing devices able to detect fully buried landmines will be added to the system, such as Ground-Penetrating-Radar (GPR). Unfortunately, the AR.drone does not have enough payload capacity for carrying a GPR, however, the detection ROS package presented in this paper could be easily extended to other drone platforms with more payload. Our mid-term goal is to embed these sensing technologies onboard a quadrotor with larger payload capacity and to provide a safer and ready-to-use autonomous system that can be easily adapted as a complementary sensing tool within a real humanitarian demining mission.

APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

The video link accompanying this paper illustrates the AR.drone covering one of the described terrains while performing autonomous detection of partially visible tuna cans acting as hand-crafted landmine-like objects.
`https://www.youtube.com/watch?v=hdfqiAXplYc`

REFERENCES

[1] H. Preetz, C. Rolf, and J. Igel, "Soil influence on landmine detectioninsights from a field study in mozambique," *Journal of Soils and Sediments*, pp. 585–605, 2013.

[2] S. Dar, Z. Tanzeel, F. Mahmood, and U. Izhar, "Gps-based landmine detection system for multiple operating units," in *Robotics and Artificial Intelligence (ICRAI), 2012 International Conference on*, 2012, pp. 45–48.

[3] K. H. Ko, G. Jang, K. Park, and K. Kim, "Gpr-based landmine detection and identification using multiple features," *International Journal of Antennas and Propagation*, 2012.

[4] L. Lazarowski and D. C. Dorman, "Explosives detection by military working dogs: Olfactory generalization from components to mixtures," *Applied Animal Behaviour Science*, pp. 84 – 93, 2014.

[5] A. Poling, B. Weetjens, C. Cox, N. W. Beyene, H. Bach, and A. Sully, "Using trained pouched rats to detect land mines: Another victory for operant conditioning," *Journal of applied behavior analysis*.

[6] I. Progrmama de Acción Integral conta Minas Antipersonal, "Situación nacional víctimas de minas antipersonales en colombia, programa presidencial para la acción integral conta minas antipersonal," 2013. [Online]. Available: http://www.accioncontraminas.gov.co/Paginas/victimas.aspx

[7] A. Celebi, M. Gullu, and S. Erturk, "Mine detection in side scan sonar images using markov random fields with brightness compensation," in *Signal Processing and Communications Applications (SIU), 2011 IEEE 19th Conference on*, 2011, pp. 916–919.

[8] J. A. GARCIA-CACERES, Rafael Guillermo; ARAOZ-DURAND and F. PALACIOS-GOMEZ, "Planning of a supply chain for anti-personal landmine disposal by means of robots," *Innovar, scieloco*, vol. 22, pp. 51 – 68, 2012.

[9] N. Shimoi and Y. Takita, "Remote mine sensing technology using a mobile wheeled robot rat-1," in *Control Automation and Systems (ICCAS), 2010 International Conference on*, 2010, pp. 622–626.

[10] N. Thnh, D. Ho, and H. Sahli, "Infrared thermography for land mine detection," in *Augmented Vision Perception in Infrared*, ser. Advances in Pattern Recognition, R. Hammoud, Ed. Springer London, 2009, pp. 3–36.

[11] Á. M. Bedoya Hernández, G. A. Guzmán Cadavid, and J. A. Chaves Osorio, "Propuesta de desarrollo robótico para el desminado humanitario," *Scientia et Technica*, vol. 3, no. 49, pp. 239–244, 2011.

[12] K. Melo, L. Paez, M. Hernandez, A. Velasco, F. Calderon, and C. Parra, "Preliminary studies on modular snake robots applied on demining tasks," in *Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)*, 2011, pp. 1–6.

[13] A. Chikwanha, S. Motepe, and R. Stopforth, "Survey and requirements for search and rescue ground and air vehicles for mining applications," in *Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference*, 2012, pp. 105–109.

[14] R. Murphy, J. Kravitz, S. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *Robotics Automation Magazine, IEEE*, vol. 16, no. 2, pp. 91–103, 2009.

[15] S. P. Kang, J. Choi, S.-B. Suh, and S. Kang, "Design of mine detection robot for korean mine field," in *Advanced Robotics and its Social Impacts (ARSO), 2010 IEEE Workshop on*, 2010, pp. 53–56.

[16] J. Lou, T. Jin, F. Liang, and Z. Zhou, "A novel prescreening method for land-mine detection in uwb sar based on feature point matching," *Geoscience and Remote Sensing, IEEE Transactions on*, pp. 3706–3714, 2013.

[17] K. Schutte, H. Sahli, D. Schrottmayer, M. Eisl, F. J. Varas, M. Bajic, M. Uppsall, and E. den Breejen, "Arc: A camcopter based mine field detection system," in *Presented at the Fifth International Airborne Remote Sensing Conference*, 2001.

[18] X. Liu, P. Chen, X. Tong, S. Liu, S. Liu, Z. Hong, L. Li, and K. Luan, "Uav-based low-altitude aerial photogrammetric application in mine areas measurement," in *Earth Observation and Remote Sensing Applications (EORSA), 2012 Second International Workshop on*, 2012.

[19] Ros - robot operating system. [Online]. Available: http://www.ros.org/

[20] PARROT.SA. (2013) Parrot ardrone 2.0. [Online]. Available: http://ardrone2.parrot.com/

[21] C. Castiblanco, J. Rodríguez, I. Mondragon, C. Parra, and J. Colorado, "Air drones for explosive landmines detection," in *ROBOT2013: First Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing. Springer International Publishing, 2014, vol. 253, pp. 107–114.

[22] M. MonajjemiEngel. (2013) ardrone autonomy ros stack. [Online]. Available: https://github.com/AutonomyLab/ardrone_autonomy/

[23] J. Engel. (2013) tum ardrone ros stack. [Online]. Available: http://wiki.ros.org/tum_ardrone

[24] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'reilly, 2008.

[25] J. Engel, J. Sturm, and D. Cremers, "Accurate figure flying with a quadrocopter using onboard visual and inertial sensing," *IMU*, vol. 320, p. 240.

[26] J. Artieda, J. Sebastian, P. Campoy, J. Correa, I. Mondragn, C. Martnez, and M. Olivares, "Visual 3-d slam from uavs," *Journal of Intelligent and Robotic Systems*, no. 4-5, 2009.

[27] R. Carnie, R. Walker, and P. Corke, "Image processing algorithms for uav "sense and avoid"," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2848–2853.

[28] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2011.

[29] J. R. Parker, *Algorithms for image processing and computer vision*. Wiley. com, 2010.

[30] C. Castiblanco, J. Rodríguez, I. Mondragon, , and J. Colorado. (2014) uav-detection. [Online]. Available: https://code.google.com/p/uav-detection/