Sensors & Transducers, Vol. 234, Issue 6, June 2019, pp. 30-36

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2019
http://www.sensorsportal.com

Towards Autonomous Investigation of Crime Scene by Using Drones

Pompílio ARAÚJO JR., Marcelo MENDONÇA, Jefferson FONTINELE and Luciano OLIVEIRA

Intelligent Vision Research Lab
Federal University of Bahia, Prof. Aristides Novis Street, 40210-630, Brazil
Tel.: +55 71 3283-9472
E-mail: pompilio.araujo@ufba.br, lrebouca@ufba.br, jefferson.font@gmail.com, eng.marcelo.mendonca@gmail.com

Received: 15 May 2019 /Accepted: 15 June 2019 /Published: 30 June 2019

Abstract: A location associated with a committed crime must be preserved, even before criminal experts start collecting and analyzing evidences. Indeed, crime scenes should be recorded with minimal human interference. In order to help specialists to accomplish this task, we propose an autonomous system for investigation of a crime scene using a drone. Our proposed autonomous system recognizes objects considered as important evidence at a crime scene, defining the trajectories through which a drone performs a detailed search. We used our previously proposed method, called Air-SSLAM, to estimate drone's pose, as well as proportional-integral-derivative controllers for aircraft stabilization. The goal is to make the drone fly through the paths defined by the objects recognized across the scene. At the end, the proposed system outputs a report containing a list of evidences, sketches, images and videos collected during the investigation. The performance of our system is assessed from a simulator, and a real-life drone system is being prepared to reach the goal.

Keywords: Criminal scene investigation, Object detection, SLAM, Autonomous drone, Self-localization.

1. Introduction

When a crime is committed, no matter how careful the thug is, evidences are spread across the crime scene, and must be collected and recorded by a team of experts. Evidences are not perennial, decreasing in quantity and quality as spatially and temporally coming far from the crime scene. The goal of collecting and recording evidences is to preserve the maximum amount of information so that experts, prosecutors and judges can analyze the dynamics of the facts, deciding in the courts on the culpability of those ones involved [1]. Gathering and recording evidences are tasks accomplished by a criminal expert, who is designated to analyze the site, as well as any material that may clarify the crime. The expert must use all technological resources available to store evidences, since fragile elements can be lost after releasing from the crime scene.

Very few works propose methods to search for crime evidence autonomously. In [2], the authors make a comparison among the ways of acquiring data from the scene with laser and images. The goal is to compare traditional methods using theodolite to geo-refer the evidence. The data collected through these 3D techniques are free of time-consuming problems and can be used at any time, while sharing between different operators; the authors propose two case studies, discussing practical aspects of data acquisition from a crime scene. In a police context, a data

acquisition system is proposed in [3] to treat environmental crimes; some aspects of data captured with aerial images are used, but there is no calculation of trajectory and the flight is carried out manually. In [4], an object detection system based on images captured from a drone equipped with an NVIDIA Jetson TX2 module for GPU processing is addressed, but does not use multiple perspectives to improve the accuracy.

Here we extend our system called AirCSI, proposed in [5]. The goal is to analyze, collect and record evidences at a crime scene from an autonomous drone. AirCSI uses a stereo camera installed on the drone to capture images used in the aircraft positioning module. This module is in charge of running AirSSLAM [6] in real-time – our method to provide simultaneous localization and mapping for drones. A downward-facing monocular camera present in the drone is used to detect and help estimate the object coordinates at the crime scene. Although AirCSI is agnostic regarding the object detection method used to find the evidences, in our experiments a YOLO-v3 [7], specially trained for our goals, was adopted. YOLO-v3 uses a single convolutional neural network to locate and classify the objects, which provides a compromise between speed and accuracy.

AirCSI brings two main contributions: 1) a method to calculate trajectories from the objects found in a crime scene; 2) a new methodology, based on multiple perspectives, to increase the accuracy of object detection. The proposed system was evaluated in a realistic environment based on the AirSim simulator [9].

2. Outline of AirCSI

Fig. 1 summarizes the proposed method, which is described in the following steps:

1) The drone initiates the movement in the vertical direction and stabilizes at the height $h < h_{max}$; Air-CSI uses the coordinate system that originates from the drone starting point;

2) Using the monocular camera at the bottom of the drone, the detected objects are classified and assigned to a relevance coefficient ρ defined by the user;

3) Trajectory calculation is performed according to the coefficient ρ of the detected evidences. A coverage radius is defined for each detected evidence, the final trajectory is the shortest path that covers the radii altogether;

4) Control module performs stabilization and displacement of the aircraft in the trajectory defined by the system. There are eight proportional-integral-derivative (PID) controllers at all. To cope with each movement direction of the quadrotor drone, four cascade loops are provided, each loop consisting of a pair of PID's aimed to control both speed and position.

5) From the data collected, AirCSI creates reports containing sketches, highlighted evidence and images gathered during the scanning.

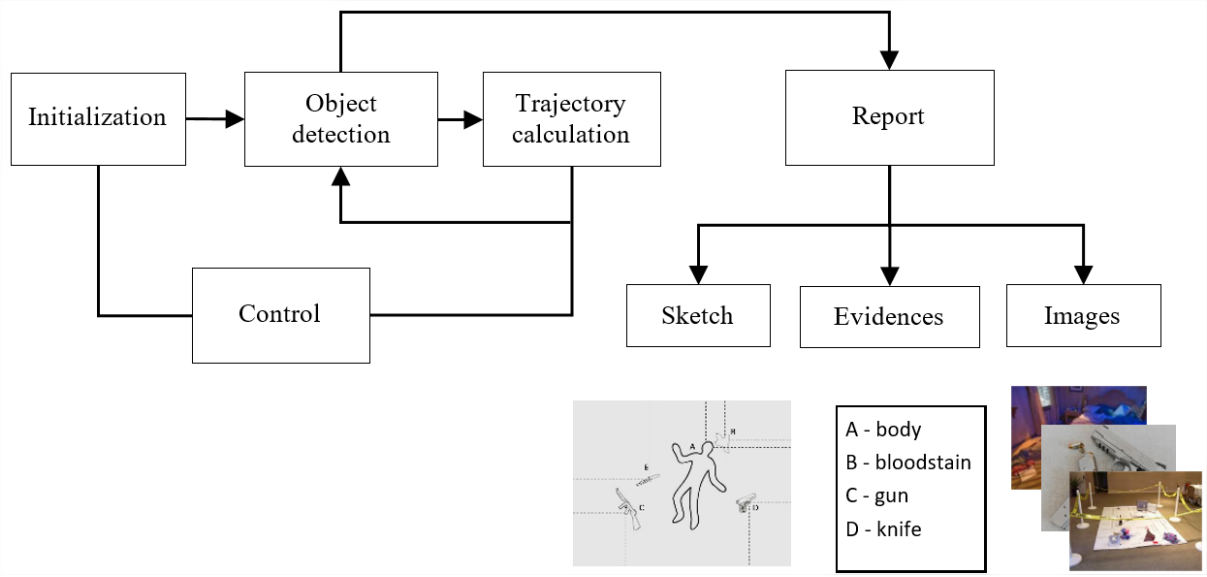

Fig. 1. Initialization - the drone takes off from a position near the crime scene, being positioned at a height *h*; **object detection** - the camera at the bottom captures images and detects suspicious objects; **trajectory calculation** - a trajectory is calculated based on the location of the detected objects, and is followed by the drone aiming to refine the object detection; **report** - the result of the scan is presented in a report.

3. Evidence Detection

YOLO was used as a baseline object detector. Although this detection method is not the most accurate, it is one of the fastest [10]. YOLO was also the best choice based on the fact that precision has less relevance than the detection rate; this is so because the object will be detected from more than one

perspective, and only objects that have their detections confirmed in all perspectives will be considered. In other words, after the first detection, object position is recorded, demanding the drone to detect the object again, in a different pose. This strategy enforces the object detection algorithm to have higher mean average precision (mAP) as the drone approaches to the object.

YOLO applies a single neural network to every whole image. The network divides the image into regions and provides bounding boxes and objectness probabilities for each region. The bounding boxes are weighted by the predicted probabilities. The most relevant classes for our training are: human body, revolver, pistol, machine gun and knife. The following parameters were used to train the object detector: batch = 64, momentum = 0.9 and decay = 0.0005. Images were preprocessed by changing their resolution to 608x608 from the original images acquired. To train the YOLO detector, MS-COCO dataset with 3000 additional weapon images were used.

After detecting the objects in the scene, each object bounding box is projected onto the ground plane, providing two dimensional information of the object location (see Fig. 2). The stereo camera is used to estimate the position of the evidence on the ground and the height of the drone. These two means also provide an estimate of the object's height.

To internally represent a detected evidence, five points of the detected bounding boxes (the four vertices and the geometric center of the rectangle) are used. Each point (p_i) in the bounding box is defined with the coordinates $X_{Ci} = [x_i\, y_i\, z_i]^T$ with respect to the camera coordinate system. Using this camera pose P, points are translated from X_{Ci} to the real world coordinate system X_{Wi}.

$$X_{Ci} = P^{-1} \cdot X_{Wi} \tag{1}$$

Each time an evidence is found, its position is stored and a counter is incremented. At the end of the scanning, the number of times each evidence was detected is recorded. The accuracy of the deviation is proportional to the value recorded by the counter, because drone overlaps the evidence many times in different perspectives (see Fig. 3).

During the scanning, the monocular camera in the drone repeatedly observe the same evidence in different perspectives. With the position of evidence recorded at the first detection, it is possible to know how many times an object was detected and its detection parameters. We then calculate a precision indicator (PI) based on the number of times the object was detected, given by

$$PI = \frac{1}{N} \sum_{i=1}^{n} C_s\,, \tag{2}$$

where C_s is the Yolo's confidence score for each image, N is the number of frames that the object should be detected and n is the number of objects detected. Then a precision indicator for each evidence is defined, taking into account all the available images of that evidence. In order to evaluate the proposed method, we consider the Intersection over Union (IoU) relative to that evidence as the average of the IoU of all images.

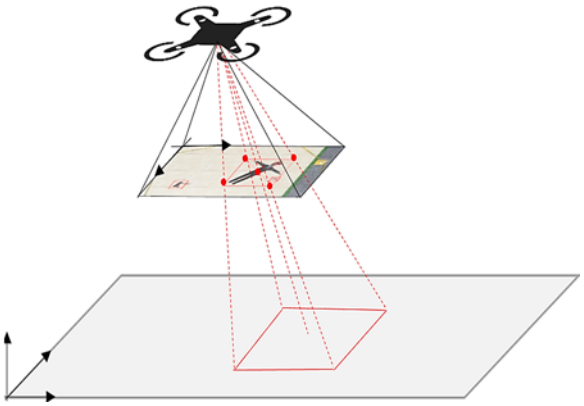

Fig. 2. The five points of the bounding box of the monocular camera image are translated to the world coordinate system by multiplying the target vector by the inverse of the pose matrix.

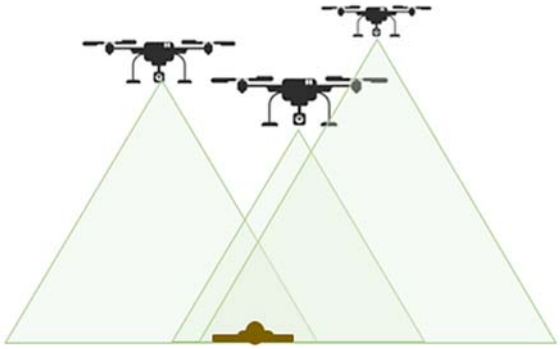

Fig. 3. Drone overlaps the evidence many times in different perspectives. The accuracy of the deviation is proportional to the number of times the drone visualizes the evidence.

4. Self-Localization

To perform the control of the drone pose in a real situation, a GPS-independent, self-locating method is exploited by using only images from the stereo camera (refer to [2] from more details about AirSSLAM).

AirSSLAM relies in good features to-track (GFTT) [11] to extract keypoints, which are lately described by rotated-binary robust-independent elementary features (rBRIEF) [12-13]. To estimate drone pose, the keypoints of the two views are matched in order to calculate the transformation matrix between two consecutive timed frames. The keypoints provide an initial map that is used as reference to be tracked posteriorly. Air-SSLAM performs a periodic map maintenance around image patches, which are also used as quality indicators to

improve keypoint tracking. An optimization procedure algorithm is applied to include new keypoints in the continuously updated map. This procedure is necessary to minimize the error between the current keypoint and the mapped (already inserted) keypoints. After that, the pose of the drone is periodically recalculated by a bundle adjustment optimization method. All mapped keypoints and the calculated pose of the drone are used to perform this refinement. Air-SSLAM presents a novel method of point matching, starting the search at a probable point location, and then gradually increasing the search area. Whenever a new keypoint is found, the likely location of the subsequent points is updated and corrected. Air-SSLAM applies a Kalman filter to stabilize the calculated camera pose. This method allows for real-time location of the drone in the environment.

AirSim is a simulator created on Unreal Engine that offers physically and visually realistic simulations designed to operate on high frequency real-time looping hardware simulations. AirSim was experimentally tested with a quadrotor as a stand-alone vehicle, comparing the software components with real-world flights. In the simulator, another computer runs the AirSim program, which transmits the pose to the drone onboard computer (NVIDIA Jetson TX2 [14]). A change was made in the original code of the *reportState* method in the AirSim in order to carry out with the transmission of the pose of the drone, via network, by using an UDP protocol. The goal was to obtain faster transmission baud-rates. This communication channel is also used to stream the controller commands to the simulator. Fig. 4 illustrates the simulation setup, and its counterpart designed to work in real situation.

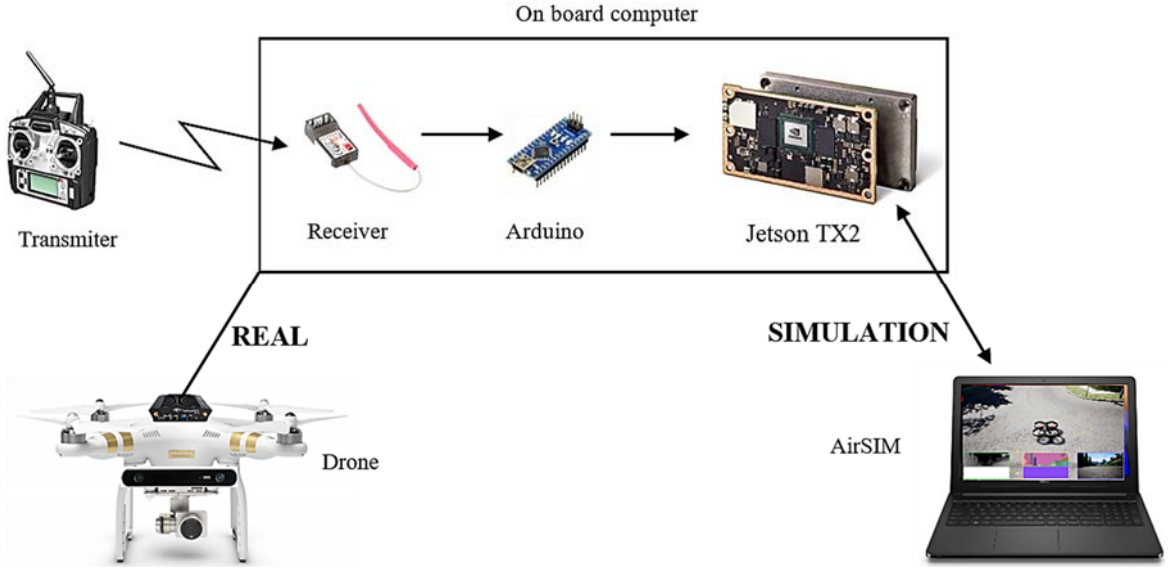

Fig. 4. Simulation tests - AirCSI runs within an NVIDIA Jetson TX2 module. Another computer runs the AirSim program, which transmits the pose to the drone's on board computer. Real projected situation - AirSSLAM and AirCSI run within an NVIDIA Jetson TX2 module. A transmitter and receiver as well as an Arduino are used for manual control.

5. Controlling the Drone

Our proposed system considers six degrees of freedom that determine the pose of the drone, $[x\ y\ z\ \varphi\ \chi\ \psi]^T$, where x, y and z are the coordinates of the drone position, and ψ is the yaw rotation. The angles φ and χ (roll and pitch) are considered null values when the drone is in equilibrium, while these values are very low during the drone movement. These constraints are completely suitable, because the drone moves at very low velocity.

A double control is applied in each direction of the drone coordinates, [x y z ψ]. For each variable, we use two controllers, one for velocity and another for position avoid accidents.

Disturbances influencing the position are corrected by the first controller, which does not reflect considerably on the velocity (see Fig. 5) [15]. In addition, the phase delay in the secondary part of the process is measurably reduced by the secondary loop. This improves the response of the velocity in the primary mesh [16].

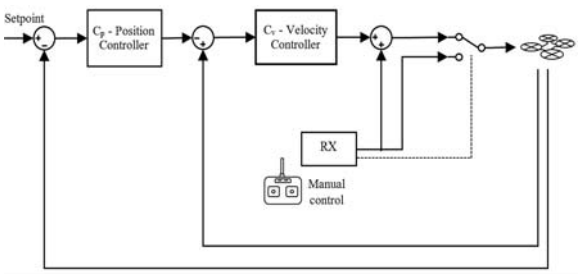

Fig. 5. Control system: the control in each direction is made by two controllers: C_V (velocity) and C_P (position). The output of the velocity controller goes into a switch that allows for the user to choose between manual or automatic control.

The input of the controller C_V is the velocity error $e_{\dot{x}_c}$, given by

$$e_{\dot{x}_c} = \left(\frac{x_{c(n)} - x_{c(n-1)}}{T}\right) - P^{-1}\dot{x}_{ws}\,, \qquad (3)$$

where $x_{c(n)}$ and $x_{c(n-1)}$ is the position of the drone in the camera coordinate system in the current and previous samplings, respectively; T is the sampling period, P is the pose matrix of the drone, and $\dot{x}_{ws}$ is the reference velocity in the global coordinate system that is received from the position controller output.

The input of the controller C_P is the position error e_{x_w}, which is defined as

$$e_{X_w} = X_w - X_{ws}\,, \qquad (4)$$

where X_w is the position of the current drone, and X_{ws} is the desired position. The PID controllers embody the transfer function

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{de}{dt}, \qquad (5)$$

where K_p, K_i, K_d are the proportional, derivative integral constants, respectively. The method 2p2z was implemented with sampling period of 160 ms. The output is given by

$$y[n] = e[n]b_0 + e[n-1]b_1 + e[n-2]b_2 + y[n-1], \qquad (6)$$

where y[n] is the control signal at the output of the controller, and e[n] is the error in the controlled variable (position or velocity). The constants b_0, b_1 and b_2 are

$$b_0 = K_p + \frac{K_i \cdot T}{2} + \frac{K_d}{T}\,,$$

$$b_1 = K_p + \frac{K_i \cdot T}{2} - \frac{2 \cdot K_d}{T}\,, \qquad (7)$$

$$b_2 = \frac{K_d}{T}$$

The controllers were tuned by the Ziegler-Nichols closed-loop method. The values of the constants for both controllers are listed in Table 1. In our experiments, a remote control is used to manually control the drone during the tuning phase.

Table 1. Coefficients of the controllers used in the tests.

	x		***y***		***z***		***ψ***	
	Cp	***Cv***	***Cp***	***Cv***	***Cp***	***Cv***	***Cp***	***Cv***
Kp	5	30	5	30	10	40	10	20
Ki	1	2	1	2	2	2	5	5
Kd	4	3	4	3	3	3	0.08	0.01

A switch key was implemented to permit switching from manual to automatic control. A routine was implemented to keep the drone in the current position whenever the key is triggered. This allows the user to set an initial pose for the drone manually.

Fig. 6 illustrates the results of the tests with the controllers running in the simulator. The controllers were evaluated by adjusting the set point with a variation of eight meters in each direction *x*, *y* and *z*, and a variation of 90º in the angle yaw (ψ). An average accommodation time of 20 s in all directions was obtained. With respect to the angle yaw (ψ), the accommodation time was 10 s. This time of 30 s can be considered as satisfactory, because a small value of velocity is needed to give room to perform the detection of objects.

A large overshoot was observed only in the *x* direction. The main reason might not have to do with the system itself, but to some delay in transmitting the position over the Ethernet network. This hypothesis was confirmed when the tests were repeated, and the problem did not occur again.

To stabilize the calculated camera pose, a Kalman filter was applied only on the position controllers, using the following parameters: Number of states = 2, measure states = 1, and time of measurement is 0.500 s.

6. Trajectory Calculation

After detecting the first evidence from a height h, AirCSI flies down to perform a detailed search for more evidences. Each type of evidence has a span radius value ρ_i that defines a scan area. In our experiments, the following radii were used for some evidence samples: human body (ρ_1 = 3 m), revolver (ρ_2 = 2 m), pistol (ρ_3 = 2 m), machine gun (ρ_4 = 2 m) and knife (ρ_5 = 1 m).

Scanning is performed in-line by following a path that fills the rectangle R that circumscribes the circle of radius ρ_i (see Fig. 7). That rectangle is calculated using the method described in [17]. This method builds a rectangle of minimum area enclosing an n-vertex convex polygon. To define the in-line scan, the drone moves following a zig-zag path, according to a sequence of points, which is defined as follows: Let V_1, V_2, V_3 and V_4 be the vertices of the rectangle circumscribing the circles, h the height of the drone during scanning and θ the horizontal aperture angle of the camera, the trajectory follows the points T_i defined as

$$T_i = V_t + \left(\frac{1}{2} + j\right) \cdot \left(\frac{V_{t+2} - V_t}{\|V_{t+2} - V_t\|}\right) \cdot h \cdot tan\,\frac{\theta}{2}\,, \qquad (8)$$

where i is the index of each point of the trajectory j

$$j = (i-1)\ div\ 2,$$
$$t = 1 + [(i-1)\ div\ 2]\ mod\ 2 \qquad (9)$$

With this trajectory, the drone is able to sweep all areas close to the evidence, guaranteeing that no point will escape from the vision of the camera. The example in Fig. 7 shows a trajectory calculated for a crime scene where four evidences were found: A human body, a revolver, a knife and a machine gun.

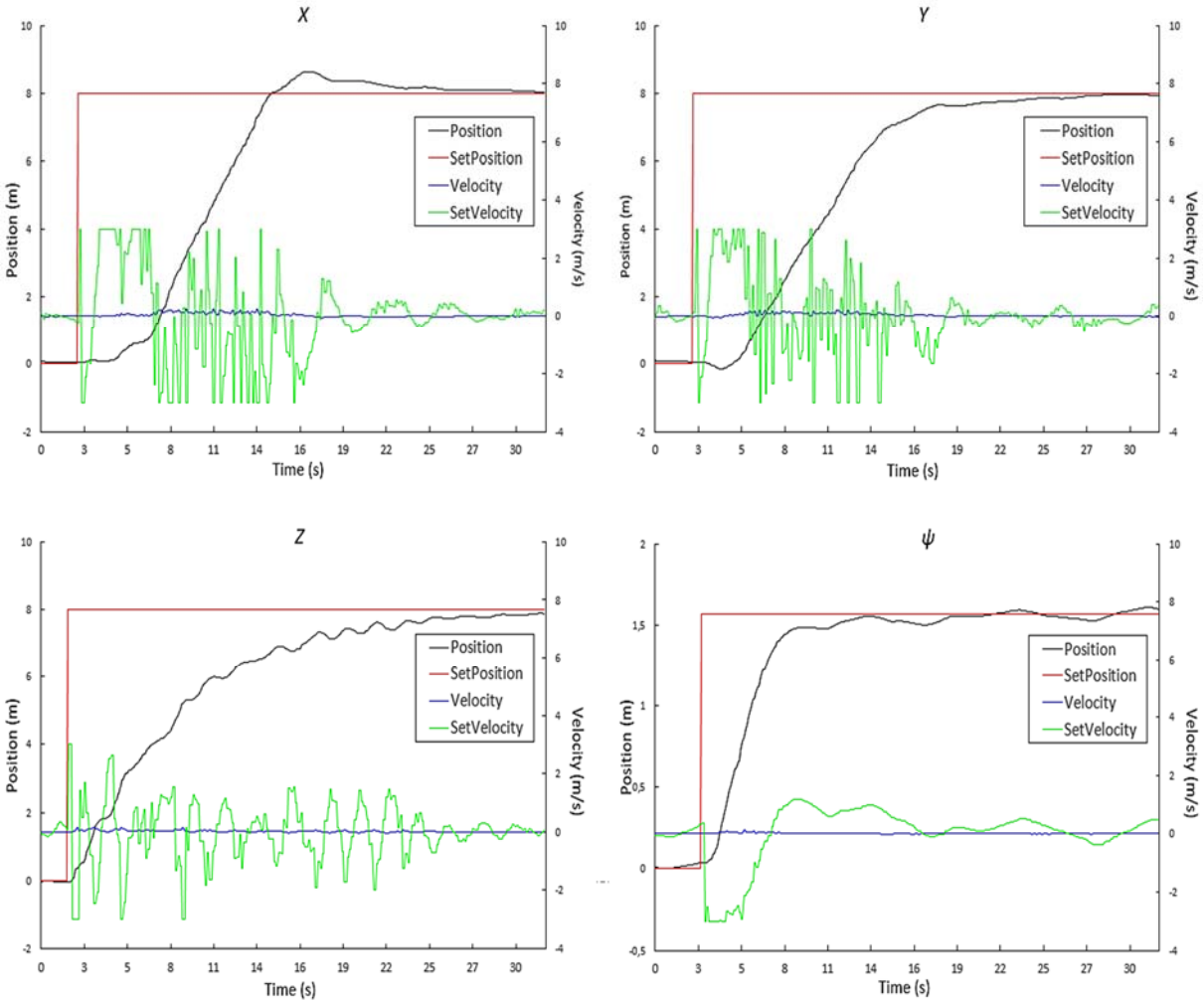

Fig. 6. Position and velocity variation over a period of 30 seconds. The average accommodation time of 20 seconds was observed in the directions x, y, x, and 10 seconds in the angle ψ.

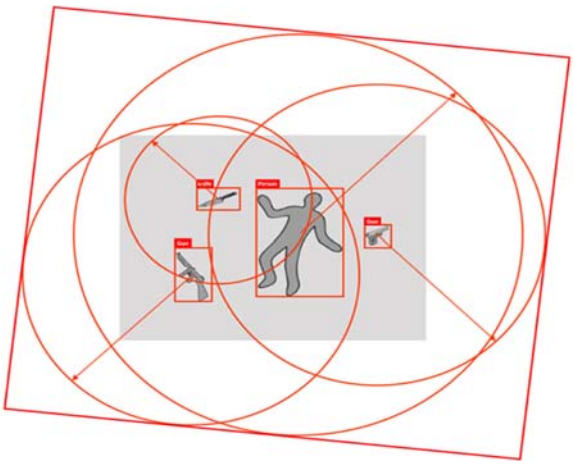

Fig. 7. Drone trajectory calculated as a function of the weight coefficient ρ_i for each evidence. The human body is more relevant than other evidences, so it has a larger scanning area.

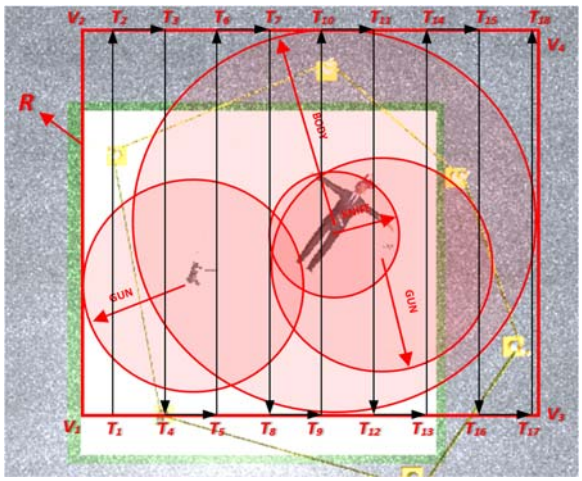

Fig. 8. Drone moves following a zig-zag path, according to a sequence of points.

7. Conclusions

AirCSI was evaluated to scan an area containing evidence of a crime, outputting a report with information on the geometry of the scene. The detection method used in the system, YOLO v3, presented 38.5 % mAP50, with real-time detection of 3 fps. With the system of multiple perspectives, we were able to improve the mAP50 to 49.6 %. We consider that in the search for evidence of crime, a low false negative value is necessary, since a human analysis will always be done by a specialist after the automatic search. The system control is designed for the drone to move steadily. The cascade controllers allowed for an explicit speed adjustment, which gave a very fast scrolling to prevent loosing image frames. Thus, the accommodation time found of 20 s can be considered adequate for scanning the tested distance of 8 m. As suggestion for future research, we are working on an actual AirCSI implementation on a drone. In addition, the goal of our research is to train a detector to look for other classes of evidence, such as ammunition, projectiles, bloodstains, and other types of objects found in crime scene. We also intend to improve the detection method in multiple perspectives and present results that are more robust.

Acknowledgements

The authors would like to thank FUNDAÇÃO DE AUXÍLIO À PESQUISA DO ESTADO DA BAHIA (FAPESB) to support with the project under grant APP0015/2016, and the Federal Police of Brazil, for the time granted to the project. Luciano Oliveira has a research scholarship from CNPq Proc. No. 307550/2018-4.

References

[1]. Fish J. T., Miller L. S., Braswell M. C., E. W. Wallace Jr, Crime scene investigation, *Routledge,* 2013.

[2]. Bucheli S. R., Pan Z., Glennie C. L., Lynne A. M., Haarman D. P., Hill J. M., Terrestrial laser scanning to model sunlight irradiance on cadavers under conditions of natural decomposition, *International Journal of Legal Medicine*, Vol. 128, Issue 4, 2014, pp. 725-732.

[3]. Lega M., Ferrara C., Persechino G., Bishop P., Remote sensing in environmental police investigations: aerial platforms and an innovative application of thermography to detect several illegal activities, *Environmental Monitoring and Assessment*, Vol. 186, Issue 12, 2014, pp. 8291-8301.

[4]. Tijtgat N., Van Ranst W., Volckaert B., Goedemé T., De Turck F., Embedded real-time object detection for a UAV warning system, in *Proceedings of the International Conference on Computer Vision (ICCV' 2017),* 2017, pp. 2110-2118.

[5]. Araújo P., Mendonça M., Oliveira L., AirCSI – Remotely Criminal Investigator, in *Proceedings of the 1st International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI'19),* Barcelona, Spain, 2019, pp. 58-63.

[6]. Araújo P., Miranda R., Carmo D., Alves R., Oliveira L., Air-SSLAM: A visual stereo indoor slam for aerial quadrotors, *IEEE Geoscience and Remote Sensing Letters*, Vol. 14, Issue 9, 2017, pp. 1643–1647.

[7]. Redmon J., Farhadi A., YOLOv3: An incremental improvement, *arXiv preprint arXiv:1804.02767*, 2018.

[8]. Shah S., Dey D., Lovett C., Kapoor A., AirSim: High-fidelity visual and physical simulation for autonomous vehicles, in *Field and Service Robotics*, 2017, pp. 621-635. https://arxiv.org/abs/1705.05065.arXiv:arXiv:1705.05066.

[9]. J. Shi, C. Tomasi, Good features to track, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593–600.

[10]. M. B. Jensen, K. Nasrollahi, T. B. Moeslund, Evaluating state-of-the-art object detector on challenging traffic light data, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 9-15.

[11]. J. Shi, C. Tomasi, Good features to track, *Cornell University*, 1993.

[12]. E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in *Proceedings of the International Conference on Computer Vision (ICCV'11)*, 2011, pp. 2564–2571.

[13]. M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010, pp. 778–792.

[14]. NVIDIA Jetson TX2, (http://www.nvidia.com).

[15]. Saha P. K., Process Control and Instrumentation-Web course.

[16]. Marlin T. E., Marlin T. E., Process control: designing processes and control systems for dynamic performance, *New York: McGraw-Hill*, Vol. 2, 1995.

[17]. Arnon D. S., Gieselmann J. P., A linear time algorithm for the minimum area rectangle enclosing a convex polygon, *Department of Computer Science Technical Reports,* 1983.