Multi-perspective object detection for remote criminal analysis using drones

Pompílio Araújo, Jefferson Fontinele and Luciano Oliveira

***Abstract*—When a crime is committed, the associated site must be preserved and reviewed by a criminal expert. Some tools are commonly used to ensure the total registration of the crime scene with minimal human interference. As a novel tool, we propose here an intelligent system that remotely recognizes and localizes objects considered as important evidences at a crime scene. Starting from a general viewpoint of the scene, a drone system defines trajectories through which the aerial vehicle performs a detailed search to record evidences. A multi-perspective detection approach is introduced by analyzing several images of the same object in order to improve the reliability of the object recognition. To our knowledge, it is the first work on remote autonomous sensing of crime scenes. Experiments showed an accuracy increase of 18.2 percentage points, when using multi-perspective detection.**

***Index Terms*—criminal scene investigation, multi-perspective object detection, intelligent drones, SLAM.**

I. Introduction

In a scene where a crime is committed, evidences are scattered nearby, and should be recorded and collected by a team of experts [1]. Evidence is not perennial, decreasing in quantity and quality over the time. 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 court the culpability of those involved. Tools to automatize the process of collecting evidences is essential to speed up the time to solve crimes, with minimal human interference [2].

In this work, a novel method to increase the accuracy of object detection based on multiple perspectives is introduced as a tool for automatic detection of objects in a crime scene. To do that, we use our AirCSI system [3] in a drone equipped with stereo and monocular cameras. The stereo camera is used to provide the aircraft with a global positioning system in real time by exploiting our simultaneous localization and mapping method (AirSSLAM) [4]. In turn, the downward-facing monocular camera is used to detect and help estimating the coordinates of the objects in the scene. Although AirCSI can use any baseline object detector to recognize objects, in our experiments, Yolo-v3 [5] was specially trained for our purposes.

II. Outiline of our system

As AirCSI initiates, a coordinate system is defined to provide the drone with a starting point. Figure 1 summarizes our proposed system described in five steps, as follows:

1) **Initialization:** The drone initiates the movement in the vertical direction, stabilizing at the height $h < h_{max}$;
2) **Object detection:** Using the monocular camera at the bottom of the drone, each detected object is classified as a type of evidence, which has 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 created for each detected evidence, while the drone passes through the scene;
4) A **control module** is in charge of drone stabilization and displacement of the aircraft in the trajectories defined by the system. There are eight proportional-integral-derivative (PID) controllers: Two cascades in each direction of the quadrotor drone movements, one for velocity, and one for position;
5) **Multi-perspective detection and report:** From the object images collected by the detector during drone trajectory, the multi-perspective detection is performed in order to provide a more accurate report with the localized evidence (sketch, evidence list and evidence images).

III. Self localization and drone control

To estimate the drone pose, our Air-SSLAM system [4] is used. The keypoints of the two views from the stereo camera are matched in order to calculate the transformation matrix between two consecutive frames. So Air-SSLAM performs a periodic map maintenance around image patches, which are also used as quality indicators to improve the estimated location of the drone.

Our proposed system considers six degrees of freedom that determines the pose of the drone $[x\,y\,z\,\phi\,\chi\,\psi]^T$, where x, y and z are the coordinates of the drone position, and ϕ is the yaw rotation. The values of the angles χ and ψ are considered null 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, $[x\,y\,z\,\psi]$. For each variable, two controllers are used: One for velocity and another for position. The use of two controllers reduces interference from fast position variations, while ensures more efficient velocity control [6].

All authors are with IvisionLab, at Federal University of Bahia, Brazil (e-mails see http://ivisionlab.ufba.br). This work was supported by Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB), under grant 7594/2015, Federal Police of Brazil and CAPES (PROAP). Luciano Oliveira has a research scholarship from CNPq, Proc. No. 307550/2018-4.

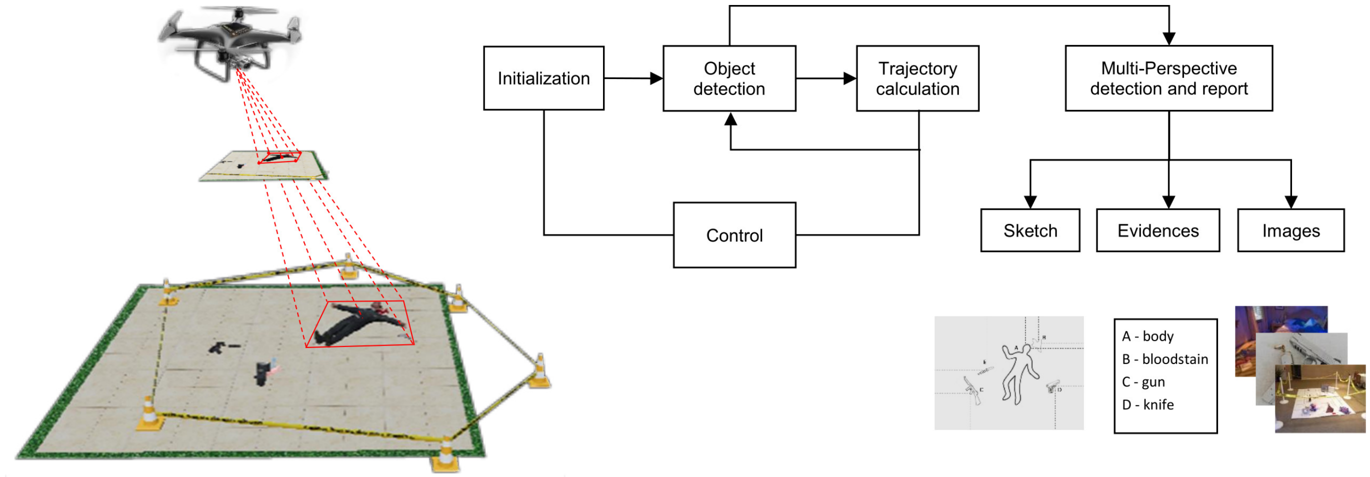


Fig. 1: **Initialization** - drone takes off from a position inside/near the crime scene, being positioned at a height h; **object detection** - the monocular camera is used to detect suspicious objects; **trajectory calculation** - with all the objects detected, a trajectory is calculated for each one of the detected objects and their locations (eventually, new objects can also be detected); **multi-perspective detection and report** - the result of the scan is presented in a report. On the left, the five points of the bounding box are translated to the world coordinate system by multiplying the target vector by the inverse of the pose matrix.

This improves the response of the velocity in the primary mesh. The input of the controller C_V is the velocity error $e_{\dot{x}_c}$, given by

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

where $x_{c(n)}$ and $x_{c(n-1)}$ are the positions of the drone in the camera coordinate system in the current and previous frames, respectively; T is the sampling period, P is the drone pose matrix 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 by

$$e_{X_W} = X_W \quad X_{WS}, \tag{2}$$

where X_W is the position of the current drone and X_{WS} is the desired position. The PID controllers are used by the transfer function:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{\mathrm{d}e}{\mathrm{d}x}, \tag{3}$$

where K_p, K_i, K_d are the proportional, derivative integral constants, respectively. The 2p2z method [7] was implemented with sampling period of $160ms$. The output is given by:

$$y[n] = e[n]b_0 + e[n \quad 1]b_1 + e[n \quad 2]b_2 + y[n \quad 1], \tag{4}$$

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:

$$\begin{aligned} b_0 &= K_p + \frac{K_i \quad T}{2} + \frac{K_d}{T}, \\ b_1 &= K_p + \frac{K_i \quad T}{2} \quad \frac{2 \quad K_d}{T}, \\ b_2 &= \frac{K_d}{T} \end{aligned} \tag{5}$$

The controllers were tuned by the Ziegler-Nichols closed-loop method [7], adjusting the set point with a variation of eight meters in each direction x, y and z, and a variation of 90° in the angle yaw (ψ). To perform experimental tests we use the AirSim simulator [8]. 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 pose information to the on-board computer (NVIDIA Jetson TX2 [9]) in the drone.

IV. Multi-perspective detection

To internally represent a detected evidence, five points (the four vertices and the geometric center of the rectangle) of the detected bounding boxes 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 to the world coordinate system X_{Wi}.

$$X_{Wi} = P^{-1} \quad X_{Ci} \; . \tag{6}$$

Each time an evidence is found, its position is stored. Its location is compared to that of all others stored. If the bounding box area matches an Intersection over Union more than 40% ($IoU > 0.4$), a counter is incremented for that evidence. At the end of the scanning, each evidence will have recorded the number of times it was detected. Then there will be more than one perspective detection for the same evidence. During scanning, the drone camera repeatedly frames the same evidence in different perspectives. With the position of

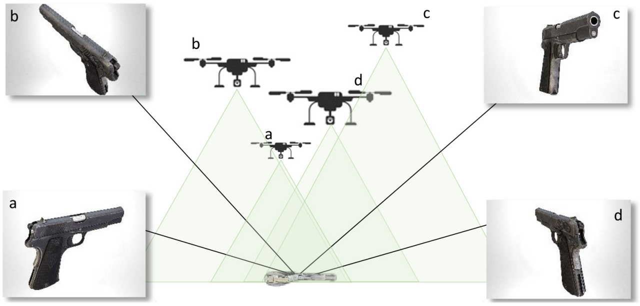


Fig. 2: Drone scans the area and objects are viewed from various perspectives. The multiple perspectives are used to improve the detection accuracy.

evidence recorded at the first detection, it is possible to know how many times an object was detected, as well as its detection parameters. A precision indicator (PI) is calculated, based on the number of times the object was detected, as follows:

$$PI = \frac{1}{N}\sum_{i=1} Cs \,, \tag{7}$$

where Cs is the confidence score of each image provided by the baseline detector, N is the number of frames that an object should be detected and n is the number of objects detected. Then a PI for each evidence is given, taking into account all the available images of that evidence. In order to evaluate the proposed method, a value of IoU is considered with respect to that evidence as the average of IoU of all images.

After detecting the objects in the scene, the object bounding box is projected onto the ground plane, providing two dimensional information of the object location (as illustrated on the left of Fig. 1).

Figure 2 shows how the drone in different positions can shows the object detected in various perspectives. Position variation allows cameras to show parts of the object that could not be viewed in a single perspective.

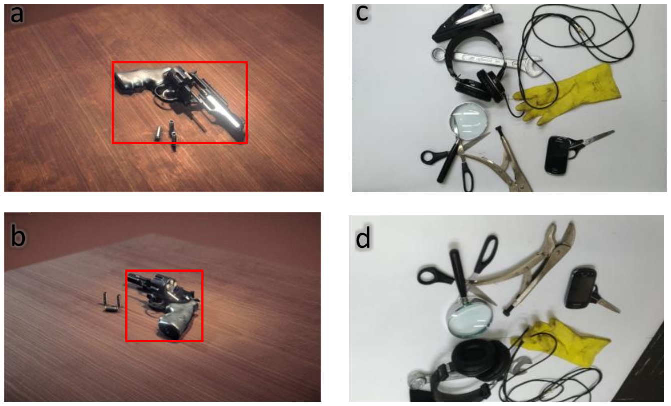


Fig. 3: (a) and (b) show two perspectives of a weapon image. (c) and (d) show two perspectives of scenes with no weapons.

TABLE I: Comparative evaluation Yolo-v3 with backbones Darknet-53 and Mobilenet-v2. Rounded average precision (in %) for AP50.

Backbone	Perspective								
	1	2	3	4	5	6	7	8	9
Darknet-53	35	43	45	47	48	50	51	52	53
Mobilenet-v2	31	37	38	39	40	41	45	46	50

A. Ablation study

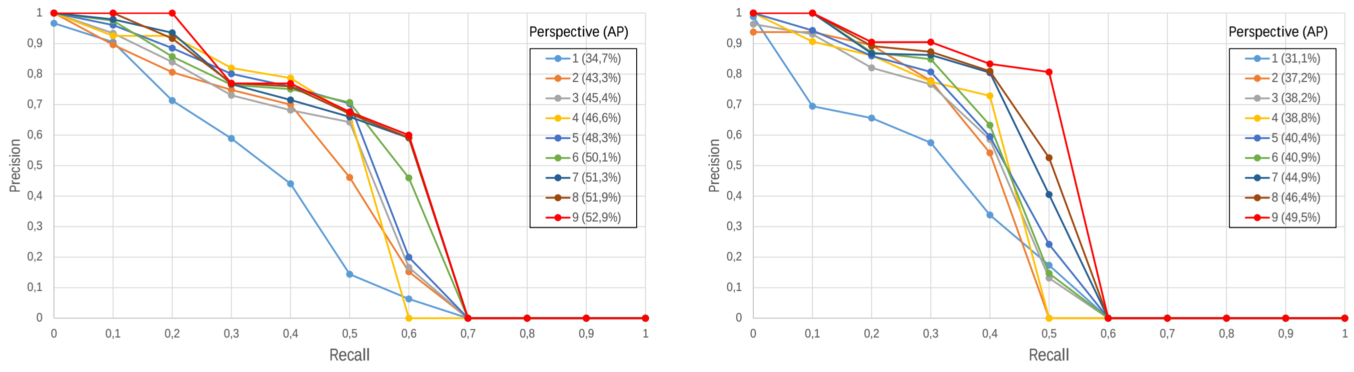
Yolo-v3 was used as a baseline detector for our multi-perspective approach. Although this detection method is not actually the most accurate nowadays, it is one of the fastest. Indeed, this issue was already shown in the work found in [5], where Yolo-v3 shows the fastest performance at the cost of presenting a lower average precision. So, Yolo-v3 was considered the best choice, since precision has less relevance than detection rate. This is so because the object will be detected from more than one perspective, and only objects that were detected more than once 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 situation makes the object detection module to have higher precision as the drone approaches to the object.

The following parameters were used to train Yolo-v3: Batch size = 64, momentum = 0.9 and decay = 0.0005. Images were preprocessed by changing their resolutions to 608 608 pixels from the original images acquired. To train the detector, MS-COCO data set [10] with 3000 additional weapon images were used [11]. In order to evaluate Yolo-v3 with different backbones, we considered the original used Darknet-53 and Mobilenet-v2 [12], both made on the Jetson TX2 machine. Table I summarizes the results found.

V. Experimental analysis

MS-COCO data set was used only for training and validation, along with the 3000 additional weapon images, allowing for a model with extensive number of categories in the future. Since there is no multi-perspective images of objects in MS-COCO data set, other 900 images containing 100 scenes in 9 different perspectives were used to test the proposed system. To evaluate the proposed system only weapon images were used housed in 50 scenes containing annotated objects, and other 50 scenes considering only objects other than weapons.

Examples of scenes containing a weapon are illustrated in Figs. 3a and 3b, while Figs. 3c and 3d depict scenes without weapons. Images were submitted to the detector individually. With the object position given by AirSSLAM, the proposed multi-perspective approach was carried out by considering the number of perspectives of each object. Considering the average precision with IoU greater than 0.5 (AP50 [13]) was possible to verify an increase of 18.2 percentage points, going from 34.7% (with one perspective) to 52.9% (with nine perspectives). Figure 4a shows the AP50 plots of the proposed multi-perspective system using Darknet-53 backbone, and Figure 4b



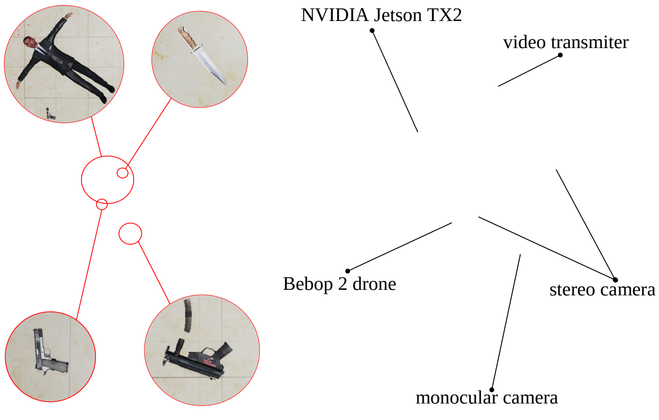
(a) (b)

Fig. 4: (a) Precision-recall curve of our proposed multi-detection system using Yolo-v3 with Darknet-53 and (b) Mobilenet-v2 backbones. In both cases the average precision AP50 increases with the number of perspectives.

illustrates the results with Mobilenet-v2 backbone. In the tests, the average detection time per image was 0.704 s for the network with the backbone Darknet-53 and 1,736 s for the network with the Mobilenet-V2 backbone. An implementation of the Mobilenet-v2 backbone found in [14] and the author's implementation of Darknet-53 [5] were used.

VI. Discussion and conclusion

Although the system proposed here uses a drone to sweep scenes with criminal evidences (particular objects), it could also be applied to monitoring difficult areas, such as archaeological parks, caves or sites covered by dense vegetation. In searching for crime evidences, a low false negative value is desired, since a human analysis will always be done by a specialist after the automatic search. In this sense, our proposed approach based on multiple perspective detection improved overall system accuracy successfully. In our experiments, a raise of 18.2 percentage points in the average precision was achieved in comparison with just one perspective. The goal is to make AirCSI autonomous to detect evidences at a crime scene (see Fig. 5a for an example, in AirSim simulator). In tests in the simulated environment, our drone were able to perform route calculation and detection of other objects, such as human bodies, knives and weapons, as well as other objects present in COCO data set. We are now working on an assembly to perform testing on a Bebop 2 type drone (see Fig. 5b). In the future, our challenge is an approach to addressing noise and occlusion of evidence.



(a) (b)

Fig. 5: (a) AirSim software simulation of a crime scene with objects detected by AirCSI. and (b) AirCSI being prepared for future testing with Parrot Bebop 2 drone.

References

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

[2] M. Lega, C. Ferrara, G. Persechino, and P. Bishop, "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, no. 12, pp. 8291–8301, 2014.

[3] P. Araújo, M. Santos, and L. Oliveira, "Aircsi remotely criminal investigator," *International Conference on Advances in Signal Processing and Artificial Intelligence*, 2019.

[4] P. Araújo, R. Miranda, D. Carmo, R. Alves, and L. Oliveira, "Air-sslam: A visual stereo indoor slam for aerial quadrotors," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 9, pp. 1643–1647, 2017.

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[6] M. Araki and H. Taguchi, "Two-degree-of-freedom pid controllers," *International Journal of Control, Automation, and Systems*, vol. 1, no. 4, pp. 401–411, 2003.

[7] T. E. Marlin, *Process control: designing processes and control systems for dynamic performance*. McGraw-Hill, 1995.

[8] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[9] "Nvidia autonomous machines," https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/, accessed: 23-March-2019.

[10] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[11] F. P. y Alberto Castillo. (2018) Weapons detection. [Online]. Available: https://sci2s.ugr.es/weapons-detection

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[14] "Mobilenet implementation," https://github.com/fsx950223/mobilenetv2-yolov3, accessed: 1-July-2019.