A Computer Vision and Control Algorithm to Follow a Human Target in a Generic Environment Using a Drone

Vitoantonio Bevilacqua(✉) and Antonio Di Maio

Department of Information and Electrical Engineering,
Polytechnic of Bari, Bari, Italy
vitoantonio.bevilacqua@poliba.it,
dimaioantonio90@gmail.com

Abstract. This work proposes an innovative technique to solve the problem of tracking and following a generic human target by a drone in a natural, possibly dark scene. The algorithm does not rely on color information but mainly on shape information, using the HOG classifier, and on local brightness information, using the optical flow algorithm. We tried to keep the algorithm as light as possible, envisioning its future application on embedded or mobile devices. After several tests, performed modeling the system as a set of SISO feedback-controlled systems and calculating the Integral Squared Error as quality indicator, we noticed that the final performance, overall satisfactory, degrades as the background complexity and the presence of disturbance sources, such as sharp edges and moving objects that cross the target, increase .

Keywords: Drones · Computer vision · Control theory · Optical flow · Classifiers · Histogram of oriented gradients

1 Introduction

The interest in using Unmanned Aerial Vehicles (UAVs) to accomplish a series of tasks that can be uncomfortable or risky to be performed by humans has been constantly increasing in the recent years for different reasons. In first place, we can notice the conspicuous increase of the number of commercial drones' sales, due to the dramatic decrease of their price and the raising amount of ways in which they can be used, ranging from games and sports to small utility applications.

This has brought to the scientific community a new interest in investigating the capabilities of UAVs [2, 3, 13, 17–21], with a particular attention to quadcopters in the consumer sector. This is because of their potential wide spread and their applications for solving generic issues of the mass consumer.

Because of these reasons, this work studies and proposes a new algorithm that makes it possible, for a commercial quadcopter with a built-in camera, to follow a walking human target relying exclusively on information extracted from the captured video stream.

D.-S. Huang et al. (Eds.): ICIC 2016, Part III, LNAI 9773, pp. 192–202, 2016.
DOI: 10.1007/978-3-319-42297-8_19

Currently, the great majority of the algorithms that accomplish similar tasks relies on color information or on using an external device to track the position of the target [7–11, 14].

For example, this link https://www.youtube.com/watch?v=JNEZmV8yONQ illustrates the outcome of a similar study, in which the algorithm tracks the target through segmentation of a blob of pixels of similar color that lie over of a piece of clothing of the target. The main drawback of this problem is also noticeable in the video: it stops working when the target is in a dark environment or when it crosses a shadow.

Different approaches use an external device attached to the target's body to track it, but it increases the overall cost and it limits the possibility to easily switch the target of the drone.

In contrast to most of the current related works, this one proposes an innovative technique to accomplish the tracking task, using no external tracking device and only the video stream coming from the drone's built-in camera. Our algorithm does not use color information, but only single-channel images extracted from the video stream frames. In case the target must be followed in total darkness, the drone can be equipped with an IR illuminator and an IR camera.

Therefore, the presented approach eliminates the drawbacks of the algorithms that solve the same problem and it is also computationally lighter, which is good when the algorithm must run on a system embedded in the drone's body.

2 Overview

This work can be divided in the resolution of three distinct sub-problems: the problem of tracking a target in a generic scene, the problem of controlling the drone and the problem of hypotheses testing.

In order to track the movement of the target in the video stream provided from the drone's camera, we need a tool that returns the position of human targets in a picture. The algorithm we have developed uses a Histogram of Oriented Gradients (HOG) classifier, which the scientific literature proved to be more successful for this specific task [6].

Unfortunately, the use of this algorithm on its own is not enough because of its relative slowness and because the video stream can reach up to 30 frames per second. The acquisition framerate is variable and depends on the speed of the algorithm: a new frame is acquired only when all the computation of the previous one has been carried out. This is why launching the classifier for each frame would decrease the frame rate to unacceptable values for controlling the drone. For this reason, the indispensable classification step has been combined with a faster tracking method, using the optical flow algorithm.

Regarding the subproblem of drone control, the intrinsic MIMO system is modeled as a set of 4 SISO systems. This is possible due to the hypothesis of independence of the components of the system state vector, which combined with the assumption of linearity, allow us to use well-known techniques of modeling and control.

For sake of simplicity, we have chosen to use proportional regulators only on 3 degrees of freedom on the total 4 controllable ones. The first and most important control is implemented on the yaw (the rotation of the aircraft around the vertical axis Z): the distance from the vertical axis of the video frame of the center of mass of the target's fiduciary points is multiplied by a multiplicative constant, to be chosen depending on the specifications of the desired response, and provided as input to the drone SDK, which in this case represents the plant. Similarly, the control is performed for the heave motion (along the vertical axis Z).

The surge motion control (along the longitudinal axis X) is performed using a more complex algorithm. The height of the target-enclosing rectangle is linked proportionally to the height of the rectangle that encloses the cloud of fiduciary points, fixed to the target's body. As a result, when the target moves away from the drone, it occurs a contraction of the points cloud for perspective effect. This contraction results in a corresponding reduction of the height of the rectangle enclosing the target and vice versa, when the points cloud expands because the target is getting closer to the drone, there is a height increase of the target box. The difference between the current height of the target box and the height we want it to reach in the video frame will be the input to the proportional controller.

From the geometric point of view, this height is proportional to the distance of the target from the drone and therefore, adjusting the coefficient of the regulator, we modify the distance from the target to which the drone will tend, as well as its reactivity to any change of the setpoint.

The evaluation of hypotheses and algorithm is performed calculating the temporal average of the ISE (Integral Squared Error) as a measure of quality. This evaluator has been chosen because it is impossible to provide canonical inputs to the system due to strong disturbances caused by various environmental and technical reasons. Therefore, the transient part of the error signal does not fade over time. This makes the classic ISE strongly dependent on the duration of the video, while a temporal average makes the evaluator less sensitive to it.

3 The Project Structure

The developed algorithm can be modeled using the following scheme, where the arrows point to the direction in which the data flows (Fig. 1).

The Acquisition thread deals mainly with the acquisition of the frame from the drone's camera and its storage on the shared global memory.

The Send Feedback Thread acquires and sends to the drone, at a fixed frequency, the values of a shared global variable containing the velocities along the three axes x, y, z and the yaw rotating speed. Actually, only 3 of 4 degrees of freedom will be controlled, because the lateral movement (y axis) will not be needed as it is redundant.

Considering the movement of the drone on a 2D horizontal plane, 2 degrees of freedom are enough to cover all of its surface. A degree of freedom we want to keep is along the x axis, because the camera is pointing along the direction of surge motion. Yaw has been chosen as second degree of freedom over the sway motion (y axis) because it changes the orientation of the x axis and therefore the direction in which the camera points. This allows the drone to follow the target always from behind and also when it walks over a curved path, things that would not be possible choosing the y axis

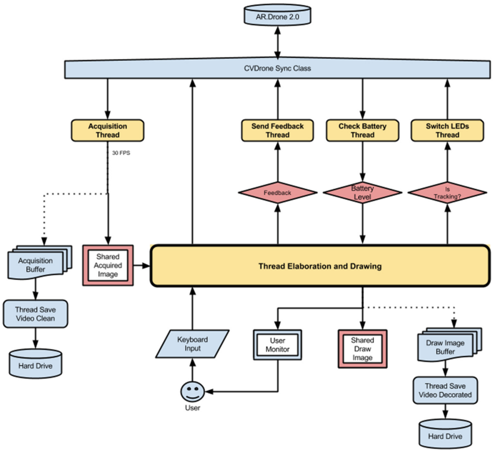

Fig. 1. General scheme of the algorithm

as second degree of freedom and therefore making the camera point constantly in the same direction.

The Elaboration and Drawing Thread is the biggest and most complex thread of the program, and it requires a particularly thorough description, which will be proposed in the next chapter.

4 The Tracking Algorithm

The elaboration thread encompasses all the algorithms to perform the core operations to accomplish the tracking task, that is HOG Classification, Good Features to Track extraction [4] and Optical Flow computation.

Hereafter, a description of the different fundamental stages through which the thread undergoes will be presented.

HOG Classification mode. When the program starts, the HOG Classifier is launched on the original frame. If the classifier does not find any target, the frame is discarded and the cycle restarts, otherwise a list of rectangles that can potentially contain the target is returned. Among all of the rectangles, the algorithm will choose the one that is closer to the last valid center of mass of the corners cloud, initially set at the frame center. This guarantees the selection of the right target in a scene with multiple ones, because when the target is lost and the classifier is relaunched, the probability to select the same target is maximum according to the principle of spatial locality.

Good features to track. After having selected the target, the Good Features to Track algorithm is launched on the area of the original image that lies under the target rectangle. If no point is returned by the algorithm, the cycle is aborted and restarted. Otherwise, the list of good points to track is stored and the center of mass is calculated.

This algorithm can operate in very cluttered environments with lots of different objects in movement. The cloud of tracking points is completely enclosed in the target box that is returned by the classification step and all the rest of the scene has no impact on the following optical flow step, considering that is an algorithm that works on local neighborhoods of the tracking points.

Optical flow tracking mode. Starting from the following cycle, and until the target is lost, the target will be tracked using the optical flow algorithm instead of the HOG classifier.

Assuming that the frame rate is high enough to keep the tracking points of the new frame in a close neighborhood of the correspondent tracking points of the previous frame, the optical flow algorithm returns a new cloud of points.

Getting rid of outliers. When the good features to track are detected on the background instead that on the foreground (the human target) they behave like outliers. The easiest way to get rid of them is to reduce as much as possible the area in which the corners will be detected [5] (virtually only on a ROI that lies perfectly over the target) and to remove the corners with a suspicious behavior.

We can locate the temporary new target box starting from the position of the old target box and translating it of the average drift vector between the two centers of mass.

All the corners that lie outside the box are discarded because they are likely to be outliers.

New Target Box Location and Dimensions. After the outliers cleaning phase, we need to adapt the dimensions of the new target box according to the actual dimensions' variation of the target. Assuming that the tracking points on the target move jointly with it, and remain bonded to the original detected points, we can say that the variation of the target box dimensions is linearly dependent on the geometrical variations of the cloud of points.

We assume that the aspect ratio of the target box is constant, so we can use just one between width and height of the bounding box of the cloud of points to modify both the target box dimensions.

In the following figure, the big red rectangle is the old target box and the small red one is the bounding box of the old cloud of points. The big blue rectangle is the new target box and the small one is the bounding box of the new cloud of points (Fig. 2).

Our goal is calculating the new value of H and W, relying upon the values of all the other variables H', h', h, W', w' and w.

Our hypothesis is that the ratio between the height of the target box and the height of the cloud bounding box is the same across two frames (the same is valid for the width), so the final formula can be written as following:

$$W = W' \frac{H}{H'} = W' \frac{h}{h'} \tag{1}$$

After calculating the two values H and W, the new target box is evenly enlarged or shrunk along all the borders by the specified quantities and stored.

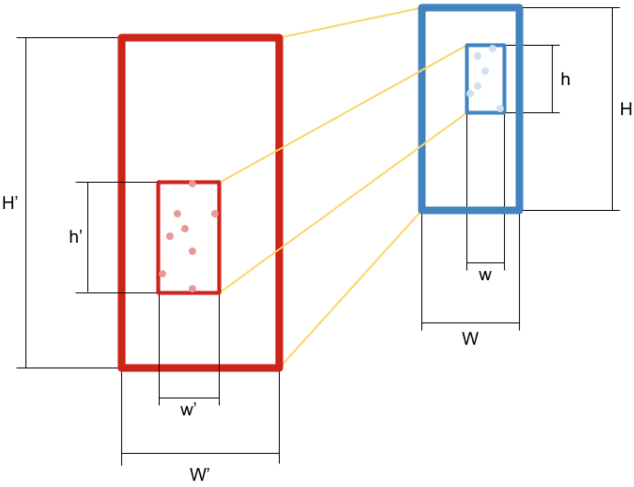

Fig. 2. The height update of the target box (Color figure online)

5 Modeling and Technical Evaluation

A method to evaluate the algorithm performance in a quantitative, numeric way is desirable. This method might be useful not only to obtain a number that expresses how good the algorithm is performing, but also a practical way to tune the algorithm parameters in order to maximize its performances, according to user's requirements.

These parameters can be considered, similarly to the well-known theories of automatic control, as coefficients of a P (proportional) controller, treating the drone as a plant with unknown transfer function G(s) or, seeing it under the digital control point of view, G(z).

The whole system can be modeled using the following classic schema:

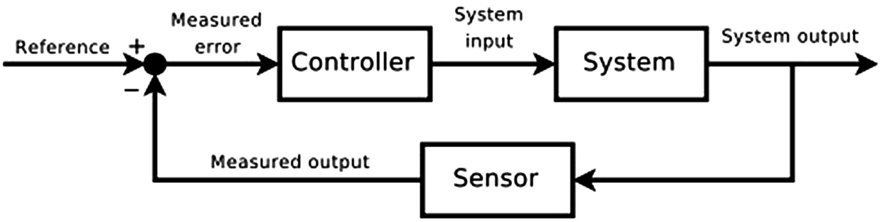

Fig. 3. The general schema of the modeled MIMO system

Actually, the whole schema can be divided in 4 similar and independent sub-schemas, focusing on each of the 4 feedback dimensions. The independence of the 4 schemas makes it much simpler to deal with the problem, because it makes a complex MIMO system, just a set of 4 easy SISO systems, which can be treated using the well-known methods from control theory. For sake of simplicity, the feedback on the y-axis, that concerns the so-called sway motion, has been kept at 0 by default. This prevents the drone from sliding side to side and it allows it only to rotate, going back and forth and up and down (Fig. 3).

In order to evaluate the performances of the control algorithm, we have used a well-known integral performance estimator, known as ISE (Integral Square Error), in which the epsilon function is the "Measured error" of the previous schema.

$$ISE = \int_{0}^{+\infty} \varepsilon^2(t)dt \tag{2}$$

For time-discrete systems, it will be enough to simply sum the square of the discrete values along the time axis. In contrast to the normal approach regarding the ISE calculation, there was no possibility to provide the system a clean and theoretical canonical input (i.e. a Heaviside step function) so the target tries to keep a steady pace when walking in the two test videos, in order to simulate a ramp function in input in the x axis system.

As well as the classic ISE, calculated on the whole data of the single test flight, an average ISE is also provided by dividing the ISE by the number of frames.

6 Testing

After having modeled the system and chosen a significant quality estimator, according to the selected model of the system, a series of practical tests have been carried out.

A brief summary of two test flights' results is shown hereafter, showing the total flight duration, graphs and a table with the ISE and Average ISE for each controlled axis.

The minimum number of tracking points for the two following test flights has been empirically set to 15. This threshold has been proven to guarantee a good compromise between tracking stability and calling the HOG classifier as less as possible (Figs. 4 and 5).

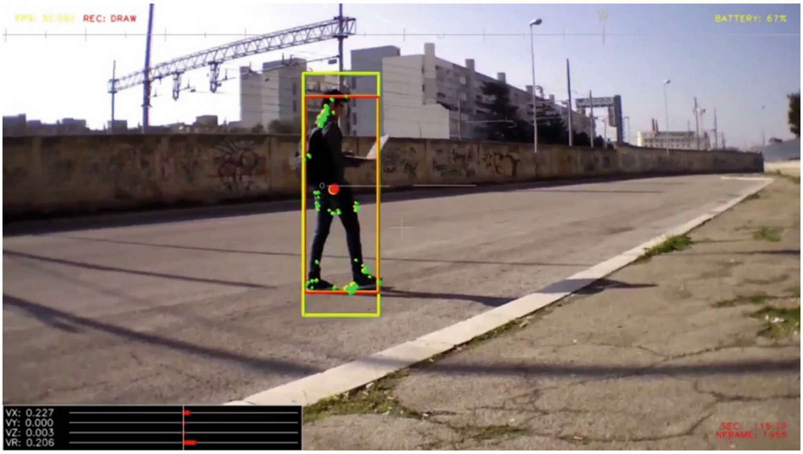

Fig. 4. The environment of test case 1 with the algorithm's graphic decorations

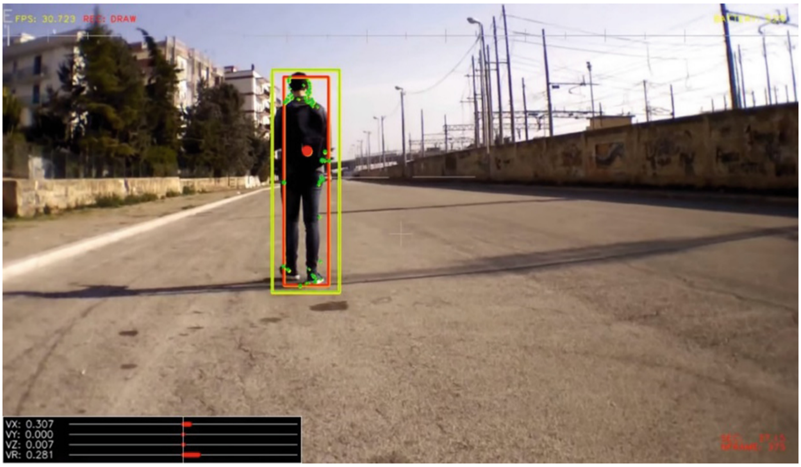

Fig. 5. The environment of test case 2 with the algorithm's graphic decorations

6.1 Test Case 1

Flight duration: 1:42

Link to the Video: https://www.youtube.com/watch?v=RzrA1blOfI4 (see Table 1 and Fig. 6)

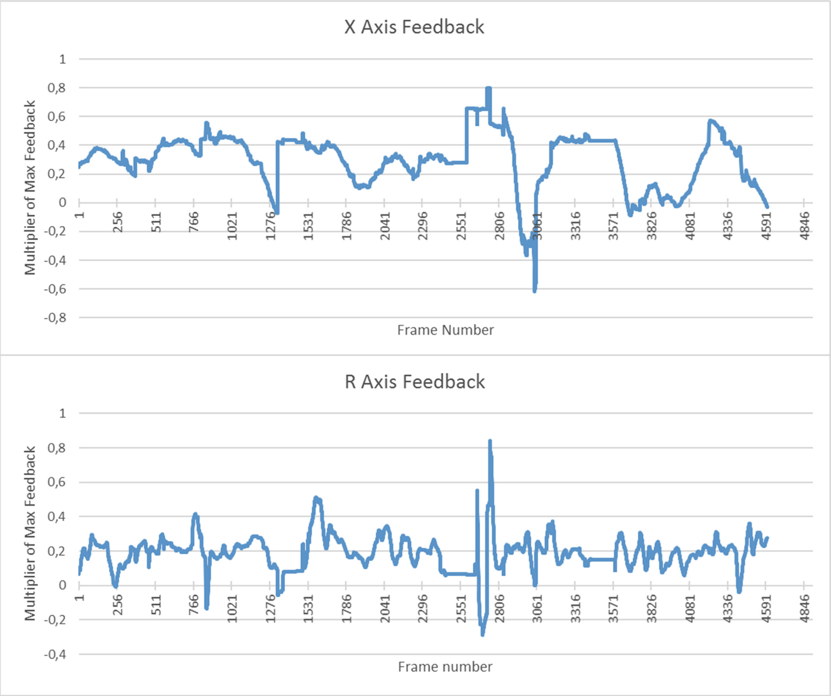

Fig. 6. The feedback graphs for the two main controlled axes (case 1)

Table 1. The performance indexes and for test case 1

AXIS	**X**	**Z**	**R**
ISE	569.660	593.993	213.534
ISE/NFRAMES	0.124	0.129	0.046

6.2 Test Case 2

Flight duration: 4:09

Link to the Video: https://www.youtube.com/watch?v=zEmeSw1BP4I (see Table 2 and Fig. 7)

Table 2. The performance indexes and the proportional regulators' factors for test case 2

AXIS	**X**	**Z**	**R**
ISE	1906.590	556.571	688.656
ISE/NFRAMES	0.179	0.052	0.065

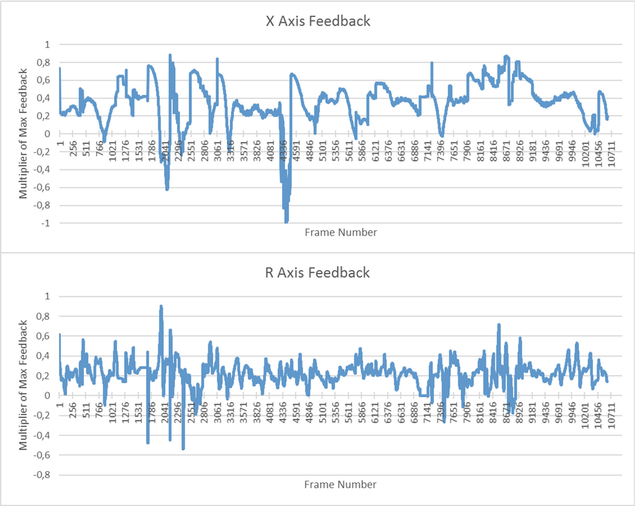

Fig. 7. The feedback graphs for the two main controlled axes (case 2)

In both of the shown test cases we can notice that the graphs have some noticeable features:

- They have some flat regions: this is due to the uneven power of Wi-Fi signal.
- They have several gaps ad leaps: this happens when the target is lost by the tracking algorithm and it must be redetected, or when the tracker is manually restarted.
- They settle, apart from the noise, steady at a certain level: this is due to the type of the system. Providing a ramp signal as input, which is a canonical input of first

order, means having a finite error of $1/|Kb|$ when the system is Type 1 (Type 1 systems have one pole in the origin. $|Kb|$ is the Bode gain).

Considering that the controller has been purposely designed as proportional, without poles in the origin, we can infer that the plant is an intrinsic low-pass filter, in particular a Type 1 system.

In order to contrast the effect of the position error and increase the static precision, it is possible to raise the Bode gain or add a pure integrator behavior to the controller in the control loop. The first strategy is unsuitable because of the bad reaction of the system to disturbances and because its dynamic response can be undesirable for specific user requirements.

Therefore, the best way to achieve disturbances rejection and static error minimization is introducing an integral action aside the proportional one. This increases the controller complexity and therefore its study will be put off to the next stages of the investigation.

7 Conclusion

In this work, a new technique for tracking a human target with a drone has been proposed and tested.

The main advantages of using this technique over the other ones are the few constraints on the environmental and target characteristics, the small computational requirements and the possibility of night tracking [15, 16], because of the independence of the algorithm from color information. Considering the lightness of the devised algorithm, one of the future expansion of the work could be embedding the algorithm in the drone's on-board card to accomplish general track-and-follow tasks. For example, as shown in [2], the system can be used to follow the position of an athlete but without the constraint of a colored shirt.

Furthermore, the target occlusion represents a problem for this as for all the other tracking algorithms. In this case, it has been solved using the variance threshold for the cloud of tracking points but it can be improved using the well-known filters such as Kalman and Particle.

In conclusion, we can say that even though the project still has a wide range of possibilities of expansion, the testing phase has shown quantitatively that the drone performs reasonably well at the task of tracking a human target in a general environment, increasing its performances as the disturbance sources decrease.

References

1. Higuchi, K., Shimada, T., Rekimoto, J.: Flying sports assistant: external visual imagery. In: AH, 12–14 March 2011
2. Nagi, J., Giusti, A., Di Caro, G.A., Gambardella, L.M.: Human control of UAVs using face pose estimates and hand gestures. In: HRI, 03–06 March 2014

3. Kos'myna, N., Tarpin-Bernard, F., Rivet, B.: Bidirectional feedback in motor imagery BCIs: learn to control a drone within 5 minutes. In: CHI, 26 April–1 May 2014
4. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
5. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference, pp. 147–151 (2001)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, vol. 1, pp. 886–893, 25 June 2005
7. Miyoshi, K., Konomura, R., Hori, K.: Above Your Hand: direct and natural interaction with aerial robot. In: ACM, 10–14 August 2014
8. Hansen, J.P., Alapetite, A., Scott MacKenzie, I., Møllenbach, E.: The use of gaze to control drones. In: ETRA, 26–28 March 2014
9. Pittman, C., LaViola Jr., J.J.: Exploring head tracked head mounted displays for first person robot teleoperation. In: IUI, 24–27 February 2014
10. Pfeil, K.P., Koh, S.L., LaViola Jr., J.J.: Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles. In: IUI, 19–22 March 2013
11. Mueller, F., Muirhead, M.: Understanding the design of a flying jogging companion. In: UIST, 05–08 October 2014
12. Mueller, F., Muirhead, M.: Jogging with a Quadcopter. In: CHI 2015 Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 2023–2032, 18 August 2015
13. Sefidgari, B.L.: Feed-back method based on image processing for detecting human body via flying robot. Int. J. Artif. Intell. Appl. (IJAIA) **4**(6), 35–44 (2013)
14. Munoz, C.A., Sobh, T.M.: Object tracking using autonomous Quad Copter. Robotics, Intelligent Sensing & Control (RISC) Lab., University of Bridgeport, 28 March 2014
15. Liang, N.S., Wan Yusoff, W.A., Dhinesh, R., Sak, J.S.: Low cost night vision system for intruder detection. In: IOP Conference Series: Materials Science and Engineering, vol. 114(1) (2016)
16. Jeong, M.R., Kwak, J.Y., Son, J.E., Ko, B., Nam, J.Y.: Fast pedestrian detection using a night vision system for safety driving. In: 2014 11th International Conference on (IEEE) Computer Graphics, Imaging and Visualization, CGIV, pp. 69–72 (2014)
17. Kimura, M., Shibasaki, R., Shao, X., Nagai, M.: Automatic extraction of moving objects from uav-borne monocular images using multi-view geometric constraints. In: IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, 12–15 August 2014
18. Chan, W.S.: Autonomous Quadcopter Flight System with Object Tracking. Department of Electronic Engineering, Undergraduate Final Year Projects, City University of Hong Kong (2015)
19. Mercado, D.A., Castillo, P., Lozano, R.: Quadrotor's trajectory tracking control using monocular vision navigation. In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 9–12 June 2015
20. Boudjit, K., Larbes, C.: Detection and implementation autonomous target tracking with a Quadrotor AR.Drone. In: 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 02, 21–23 July 2015
21. Harik, E.H.C., Guérin, F., Guinand, F., Brethé, J.F., Pelvillain, H., Zentout, A.: Vision based target tracking using an unmanned aerial vehicle. In: 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts, ARSO 2015, Lyon, France, July 2015