Efficient Camera Control using 2D Visual Information for Unmanned Aerial Vehicle-based Cinematography

Nikolaos Passalis, Anastasios Tefas and Ioannis Pitas
Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 54 124, Greece
Email: passalis@csd.auth.gr, tefas@aiia.csd.auth.gr, pitas@aiia.csd.auth.gr

***Abstract*—Using Unmanned Aerial Vehicles (UAVs), also known as *drones*, for covering public sport events, such as bicycle races, is becoming increasingly popular. Even though the problem of controlling the flight path of a drone is well studied in the literature, little work has been done on controlling the shooting camera for producing professional grade video footage. In this work we propose a fast and efficient proportional-integral-derivative (PID) based control algorithm that rely solely on 2D visual information and we demonstrate that it is possible to accurately control the camera without inferring the 3D position of the target. To ensure that the proposed method will not exhibit undesired behavior, a genetic algorithm is used to tune its parameters using a properly defined fitness function. The proposed method is evaluated using two datasets that contain actual drone footage: a dataset that contains videos of a single cyclist, and a dataset that contains actually footage from a bicycle race event, the Giro D'Italia bicycle race.**

I. Introduction

Using Unmanned Aerial Vehicles (UAVs), also known as *drones*, for covering public sport events, such as bicycle races, is becoming increasingly popular, since drones are capable of shooting spectacular videos that would otherwise very difficult and expensive to obtain. However, using a drone in a professional shooting scenario requires at least two operators, one for controlling the flight path of the drone and avoiding possible hazards, and one for controlling the shooting camera [1], [2]. The problem of controlling the flight path of a drone, using either visual or landmark/map information, is well studied in the literature [3], [4], [5], [6], [7], [8]. On the other hand, little work has been done on controlling the shooting camera of a drone for producing professional grade video footage [1], [9].

To tackle the problem of controlling the shooting camera the 3D location of the main actor (target), e.g., a cyclist, is usually used [10]. However, acquiring a *reliable* estimation of the 3D position of a target requires either a target equipped with an accurate localization device, e.g., Radio Frequency Identification (RFID) [11], and/or Global Positioning System (GPS) devices [12], or a drone equipped with sensors that can be used to infer the 3D position of the target, e.g., a LIDAR [13]. However, it is not always possible to have targets equipped with localization devices, while using extra drone equipment, such as LIDAR, increases the drone's weight and reduces its flight time. Furthemore, geometry-based camera control requires careful calibration of the resulting system as well as very accurate localization devices (especially when used for difficult and precise shots, such as close-ups).

To overcome the aforementioned limitations, we propose a fast and efficient method that is capable of accurately controlling the shooting camera of a drone using only visual 2D information. To this end, a 2D target detection [14], or tracking algorithm [15], is used to locate a target in each frame and then a proportional-integral-derivative (PID) controller is employed to appropriate control the pan, tilt and zoom of the camera according to the specification of each shot. The proposed approach is also closer to the way that humans perform camera control, i.e., we control the camera and the zoom to maintain the target into a specific part of the shot without calculating its precise position. The different dynamics of each shot are effectively handled by an appropriately tuned PID controller. Also, note that if the target is already tracked, e.g., for controlling the flight path of the drone, then the 2D position of the target will be already available. In this case, the proposed method can use this information to control the camera of the drone with minimal computational overhead, since there is no need to use a separate detection/tracking algorithm. Finally, note that other tasks, such as crowd detection [16], pose estimation [17], [18], and face detection and recognition [19], [20], might also ran simultaneously limiting the available computational resources. Thus, it is critical that a lightweight approach, that possibly exploits information that has been already extracted by other visual analysis tasks, will be used for controlling the camera.

The main contribution of this paper is the proposal of a visual information-based camera control algorithm. To the best of our knowledge, this is the first time that 2D visual information is directly used, without inferring the 3D position of the target, for controlling the camera of a drone for cinematography tasks using a fast and efficient PID-based control algorithm. A genetic algorithm is used to tune the parameters of the PID controller in order to maximize a fitness function that measures the quality of the resulting shots. That way, we ensure that the obtained PID controller will not exhibit undesired behavior, e.g., losing the targets or lagging behind them. The proposed method is evaluated using two datasets that contain drone footage: a dataset that contains videos of a single cyclist, and a dataset that contains actually footage from a bicycle race event, the Giro D'Italia bicycle race.

The rest of the paper is structured as follows. In Section II the proposed method along with the proposed tuning algorithm are presented in detail. Next, in Section III, the proposed method is evaluated using two datasets that contain actual drone footage. Finally, conclusions are drawn and future work is discussed in Section IV.

II. Proposed Method

Assume that an object detector [14], or an object tracker [15], is used to provide the minimum enclosing bounding box $\mathcal{B}$ of an object of interest, e.g., a cyclist. Let $(b_x^{(t)}, b_y^{(t)})$ be the center of the bounding box $\mathcal{B}$ at time t. The area of the box $\mathcal{B}$ can be calculated as $b_a^{(t)} = b_w^{(t)} b_h^{(t)}$, where $b_w^{(t)}$ and $b_h^{(t)}$ are the width and the height (at time t) of the bounding box respectively. To simplify the presentation of the method we assume that both the coordinates and the width/height of the bounding box are normalized in the interval $0 \dots 1$. Then, we can define the horizontal control error, the vertical control error and the zoom control error using the following vector:

$$\mathbf{e}^{(t)} = (b_x^{(t)}, b_y^{(t)}, b_a^{(t)}) - (d_x, d_y, d_a) \in \mathbb{R}^3, \quad (1)$$

where (d_x, d_y) is the target center position of $\mathcal{B}$ and d_a is the target area (frame coverage). The targets are set according to the specifications of each shot type, e.g., using the *rule of thirds* [21]. The control process is illustrated in Figure 1. The camera must be appropriately controlled in order to maintain the center of the bounding box at the specified target position (denoted by $\mathcal{D}$) as well as keep the bounding box at the specified target size by controlling the zoom.

Proportional-integral-derivative (PID) controllers are successfully used in a wide range of domains, ranging from controlling heavy industrial equipment to everyday applications, due to their ability to perform accurate control avoiding unwanted oscillations and effectively responding to disturbances [22]. In this work a PID controller is used to control the camera inputs, i.e., its pan, tilt and zoom. The output of the PID controller is defined as follows:

$$\mathbf{u}^{(t)} = \mathbf{K}_p \mathbf{e}^{(t)} + \mathbf{K}_i \int_0^t \mathbf{e}^{(\tau)} d\tau + \mathbf{K}_d \frac{d\mathbf{e}^{(t)}}{dt}, \quad (2)$$

where all the operators are applied element-wise and $\mathbf{K}_p \in \mathbb{R}^3$, $\mathbf{K}_i \in \mathbb{R}^3$, and $\mathbf{K}_d \in \mathbb{R}^3$ are the coefficients of the proportional, integral and derivatives terms. The output of the controller is a vector containing the control commands for the horizontal, vertical and zoom controls, i.e., $u^{(t)} = (u_x^{(t)}, u_y^{(t)}, u_z^{(t)})$.

Tuning the parameters of the PID controller ($\mathbf{K}_p$, $\mathbf{K}_i$ and $\mathbf{K}_d$) is not a straightforward task [23]. In this work we employ a genetic algorithm strategy [24], [25], to tune the parameters of the PID controller towards optimizing a fitness function that express the *quality* of the obtained shots. Genetic algorithms allows for obtaining solutions to difficult optimization problems using techniques, such as *crossover combination* and *mutation* of the solutions, inspired by the process of natural selection [25]. The following process was used to optimize the parameters of the PID controller:

1) An initial population of N_{init} possible parameters/solutions is created. Each parameter is represented using 16-bit floating point numbers according to the standard IEEE 754 representation [26].
2) The population is replaced by combining the fittest solutions (as evaluated using a fitness function $\mathcal{F}$) using the uniform crossover operator [25]. Mutation (random bit flips) with rate $p_m = 10^{-2}$ is also used. Furthermore, the two best solutions of the previous population are directly passed into the new population pool (elitist selection).
3) The previous step is repeated for N_{iters} times and the best solution, according to the fitness function $\mathcal{F}$, is selected.

It is evident that the quality of the obtained solution (controller) crucially depends on the way that the fitness function is defined. A simple function that measures the mean control fitness as the inverse of the absolute control error can be used:

$$\mathcal{F} = (\frac{1}{t} \int_0^t ||\mathbf{e}^{(\tau)}||_1 d\tau)^{-1}, \quad (3)$$

where $||\mathbf{x}||_1$ is the $l^{(1)}$ norm of the vector $\mathbf{x}$. However, in the conducted experiments it was established that even though maximizing Eq. (3) leads to controllers that can accurately follow the tracked objects, it can cause unwanted camera oscillations. To overcome this issue we modify Eq. (3) by including a term that measures the stability of the obtained shots. There are several different ways to define the stability of the camera control. In this work, we estimate the stability of the control by the slope of the error surface (having small error derivatives means that there is no sudden movements that can possibly lead to loosing the tracked object or degenerating the quality of the resulting video). Therefore, the fitness function is redefined as:

$$\mathcal{F} = (\frac{1}{t} \int_0^t ||\mathbf{e}^{(\tau)}||_1 d\tau + \alpha \frac{1}{t} \int_0^t ||\frac{d\mathbf{e}^{(\tau)}}{d\tau}||_1 d\tau)^{-1}, \quad (4)$$

where α is the weight of the term that measures the stability of the obtained shots. Maximizing Eq. (4) leads to a controller that can accurately follow the tracked object, without lagging behind the current object position and avoiding unnecessary oscillations. Since the designed controller operates in discrete time the fitness is estimated as follows:

$$\mathcal{F} = (\frac{1}{N} \sum_{k=0}^{N} ||\mathbf{e}^{(k)}||_1 d + \alpha \frac{1}{N} \sum_{k=1}^{N} ||\frac{\mathbf{e}^{(k)} - \mathbf{e}^{(k-1)}}{\Delta t}||_1)^{-1} \quad (5)$$

where N is the number of performed control steps and Δt is the time interval between two control steps.

III. Experimental Results

In this Section the experimental evaluation of the proposed approach is presented. First, the experimental setup is presented and then the experimental results are reported and discussed.

A. Experimental Setup

A camera control simulation environment was used for tuning the parameters of the proposed technique and evaluating its performance. To simulate the control of the camera using video streams the following process is used. First, a camera window is defined inside the input video frames. Then, the camera control commands are translated into movements of

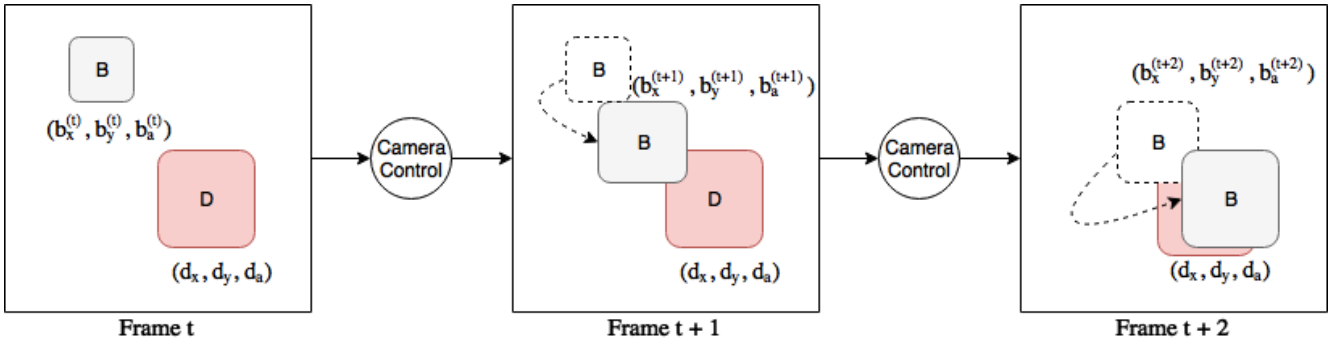


Fig. 1. Camera control using 2D visual information. The camera is appropriately controlled to ensure that the bounding box $\mathcal{B}$ reaches its target position and size $\mathcal{D}$.

TABLE I. DIFFERENT EVALUATION SETUPS

Setup	d_x	d_y	d_a	Used for tuning
Setup 1	0.5	0.5	0.5	Yes
Setup 2	0.2	0.7	0.4	Yes
Setup 3	0.7	0.2	0.2	Yes
Setup 4	0.4	0.6	0.6	No
Setup 5	0.1	0.4	0.1	No

this window, e.g., moving the camera left would appropriately move the window to the left, while zooming the camera would make the window smaller. The left/right/up/down camera movements cause an additive translation of the window, while zooming in/out cause a multiplicative translation, i.e., the size of the window is translated as $w_a^{(t+1)} = w_a^{(t)}(1+u_z^{(t)})^2$, where $w_a^{(t)}$ is the area covered by the window in the t-th frame and $u_z^{(t)}$ the output of the controller.

Two datasets were used for the conducted experiments. The first one, called *DRONE-S* in this work, is a dataset composed of 3,251 frames of annotated drone footage of a single cyclist in various settings collected from YouTube. The training set, used for tuning the PID controller, is composed of a small part of the dataset (510 frames), while the rest of the dataset (2,741 frames) are used to evaluate the performance of the algorithm. The other dataset, called *GIRO-D*, is composed of 5,000 frames of actual footage of the Giro D' Italia bicycle race, where either one individual cyclist is tracked or a crowd of cyclists (if the drone is far away from the cyclists).

To evaluate the proposed technique under a wide range of shooting scenarios five different evaluation setups were used. For each of them a different target $\mathcal{D}$ was used. Table I summarizes the used evaluation setups. Note that only three of them (Setup 1, 2 and 3) were used for tuning the PID controller, while all of them were used to evaluate its performance. Using two additional evaluation setups in the testing allows for evaluating the performance of the controller on scenarios that were not seen during the tuning process.

For the genetic optimization algorithm a pool of 50 solutions was used and the optimization ran for 50 iterations. The initial random pool was generated using Gaussian noise $\mathcal{N}(\mu, \sigma)$, where μ is the mean and σ is the standard deviation. Note that initializing the solution pool into a relatively good region of the solution space can significantly accelerate the convergence of genetic algorithms. Therefore, the initial random solutions for the parameter $\mathbf{K}_p$ were generated using $\mathcal{N}(10^{-2}, 10^{-2})$, while $\mathcal{N}(10^{-4}, 10^{-4})$ was used for generating the solutions for the $\mathbf{K}_i$ and $\mathbf{K}_d$ parameters. It was experimentally established that solutions in this region provide qualitative good control results. The weighting parameter of the fitness function was set to $\alpha = 0.5$ for all the conducted experiments. Using larger values for α allows for learning controllers that have exhibit more stable behavior. However, this can decrease the ability of the controller to track fast moving targets.

TABLE II. DRONE-S: EVALUATION RESULTS

The weighted control error $\mathcal{E}$ is reported for the five different evaluation setups.

Method	Setup 1	Setup 2	Setup 3	Setup 4	Setup 5
Baseline	0.735	0.685	0.485	0.822	0.385
Best Random	0.616	0.571	0.382	0.709	0.323
Optimized	**0.547**	**0.475**	**0.296**	**0.651**	**0.242**

TABLE III. GIRO-D: EVALUATION RESULTS

The weighted control error $\mathcal{E}$ is reported for the five different evaluation setups.

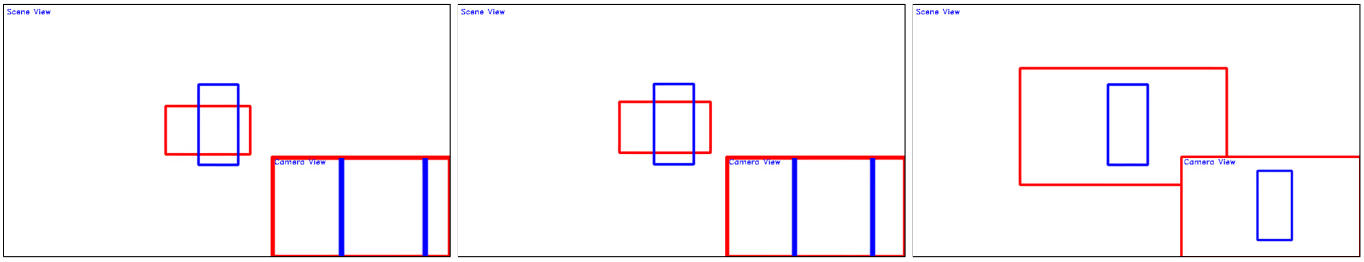
Method	Setup 1	Setup 2	Setup 3	Setup 4	Setup 5
Baseline	0.746	0.666	0.734	0.695	0.736
Best Random	0.737	0.659	0.696	0.711	0.753
Optimized	**0.702**	**0.550**	**0.644**	**0.667**	**0.693**

B. Experimental Evaluation

First, the evaluation results using the test split of the DRONE-S dataset for the five different evaluation setups are reported in Table II. The weighted sum of the control and stability error, i.e.,

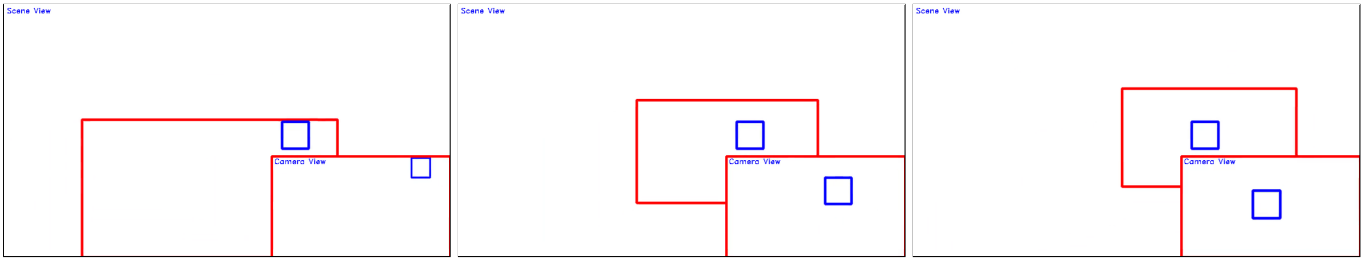
$$\mathcal{E} = \frac{1}{N}\sum_{k=0}^{N} ||\mathbf{e}^{(k)}||_1 d + \alpha \frac{1}{N}\sum_{k=1}^{N} ||\frac{\mathbf{e}^{(k)} - \mathbf{e}^{(k-1)}}{\Delta t}||_1 \quad (6)$$

is reported. The proposed optimization technique is compared to both the best solution found in the initial pool of solutions (called "Best Random"), as well as to the baseline solution selected by a few visual experiments (called "Baseline", $\mathbf{K}_p = (10^{-2}, 10^{-2}, 10^{-2}1)$ and $\mathbf{K}_i = \mathbf{K}_d = (10^{-4}, 10^{-4}, 10^{-4})$). The proposed optimization technique significantly reduces the



Baseline Best Random Optimized

Fig. 2. Sample Frame 1: Comparison between the "Baseline" (left), "Best Random" (center) and "Optimized" (right) solutions. The red box corresponds to the camera, while the blue box to the tracked cyclist. The actual video frame was removed to comply with the video license. (Figure best viewed in color)



Baseline Best Random Optimized

Fig. 3. Sample Frame 2: Comparison between the "Baseline" (left), "Best Random" (center) and "Optimized" (right) solutions. The red box corresponds to the camera, while the blue box to the tracked cyclist. The actual video frame was removed to comply with the video license. (Figure best viewed in color)

weighted sum of the control and stability error over both the baseline and the best random solution for all the evaluated setups. The evaluation results for the GIRO-D dataset are shown in Table III. Again, using the proposed technique to tune the PID controller leads better results, significantly reducing the total error for the evaluated setups.

We have also performed a qualitative analysis to validate the improved performance of the tuned controller. Two sample frames where the three controllers are compared are shown in Figures 2 and 3. Note that the actual frame of the video was removed to comply with the video license. The red box corresponds to the camera, while the blue box to the tracked cyclist. The target was set to the center of the camera, i.e., $(d_x, d_y) = (0.5, 0.5)$, while the target area was set to 0.2. The first two non-optimized controllers are unable to promptly respond to sudden movements and appropriately control the camera. On the other hand, the tuned controller is able to accurately track the cyclist. We have also observed that the optimized controller can sometimes respond more slowly to zoom changes. However, this behavior seems especially important to avoid loosing the target and provides smoother and more stable videos.

IV. Conclusion

In this work a method for controlling the shooting camera of a drone using solely 2D visual information was proposed. After detecting and tracking the target, a PID controller was used to issue the appropriate commands to the camera. Hence we demonstrated that it is possible to perform purely visual camera control, in a way similar to the way humans control the camera, avoiding the need for accurately localizing the target. A genetic algorithm was used to tune the parameters of the controller and ensure that the tracked object is accurately followed by the camera, without lagging behind the current object position and avoiding unnecessary oscillations. Two datasets, including one dataset that contains footage from a real cyclist event, were used to evaluate the proposed method and demonstrate the ability of the proposed approach to effectively control the camera and provide professional grade video footage.

There are several interesting future work directions. First, more advanced "cinematography"-oriented fitness functions can be defined to allow for tuning the controller towards more specific shot requirements. Furthermore, since different shots might require different behavior from the controller, the parameters can be replaced on demand according to the requested shot type (similar to *gain scheduling* strategies [27]). Finally, optimal control techniques, such as *reinforcement learning* [28], can be also used in a similar manner to control the camera using only 2D visual information. Neural PID controllers approaches [29], possibly combined with compact representations that carry information extracted from the current frame [30], could be also used for learning a lightweight neural PID controller end-to-end.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Transactions on Graphics*.

[2] N. Passalis and A. Tefas, "Concept detection and face pose estimation using lightweight convolutional neural networks for steering drone video shooting," in *Proceedings of the 25th European Signal Processing Conference*, 2017.

[3] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, "A vision-based guidance system for uav navigation and safe landing using natural landmarks," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 233, 2010.

[4] S. Leven, J.-C. Zufferey, and D. Floreano, "A minimalist control strategy for small uavs," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2009, pp. 2873–2878.

[5] Z. He, R. V. Iyer, and P. R. Chandler, "Vision-based uav flight control and obstacle avoidance," in *Proceedings of the American Control Conference*, 2006, p. 5.

[6] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying uav," in *Proceedings of the International Conference Intelligent on Robots and Systems*, 2011, pp. 4929–4934.

[7] A. Qadir, J. Neubert, W. Semke, and R. Schultz, "On-board visual tracking with unmanned aircraft system (uas)," in *Proceedings of the AIAA Infotech at Aerospace Conference and Exhibit*, 2011.

[8] W. Si, H. She, and Z. Wang, "Fuzzy pid controller for uav tracking moving target," in *Proceedings of the Control And Decision Conference*, 2017, pp. 3023–3027.

[9] H. Zou, Z. Gong, S. Xie, and W. Ding, "A pan-tilt camera control system of uav visual tracking based on biomimetic eye," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 1477–1482.

[10] S. Sohn, B. Lee, J. Kim, and C. Kee, "Vision-based real-time target localization for single-antenna gps-guided uav," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1391–1401, 2008.

[11] K. Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010.

[12] A. K. Brown and M. A. Sturza, "Vehicle tracking system employing global positioning system (gps) satellites," Jul. 6 1993, uS Patent 5,225,842.

[13] S. E. Reutebuch, H.-E. Andersen, and R. J. McGaughey, "Light detection and ranging (lidar): an emerging tool for multiple resource inventory," *Journal of Forestry*, vol. 103, no. 6, pp. 286–292, 2005.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[15] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," in *Proceedings of the Scandinavian Conference on Image Analysis*, 2013, pp. 652–663.

[16] M. Tzelepi and A. Tefas, "Human crowd detection for drone flight safety using convolutional neural networks," in *Proceedings of the European Signal Processing Conference*, 2017, pp. 743–747.

[17] N. Passalis and A. Tefas, "Improving face pose estimation using long-term temporal averaging for stochastic optimization," in *Proceedings of the International Conference on Engineering Applications of Neural Networks*, 2017, pp. 194–204.

[18] ——, "Learning bag-of-features pooling for deep convolutional neural networks," in *Proceedings of the International Conference on Computer Vision*, 2017.

[19] D. Triantafyllidou, P. Nousi, and A. Tefas, "Lightweight two-stream convolutional face detection," in *Proceedings of the European Signal Processing Conference*, 2017, pp. 1190–1194.

[20] P. Nousi and A. Tefas, "Discriminatively trained autoencoders for fast and accurate face recognition," in *Proceedings of the International Conference on Engineering Applications of Neural Networks*, 2017, pp. 205–215.

[21] D. Rowse, "Rule of thirds," *Digital Photography School*, 2012.

[22] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

[23] A. O'Dwyer, *Handbook of PI and PID controller tuning rules*. World Scientific, 2009.

[24] C. Li and J. Lian, "The application of immune genetic algorithm in pid parameter optimization for level control system," in *Proceedings of the IEEE International Conference on Automation and Logistics*, 2007, pp. 782–786.

[25] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*. John Wiley & Sons, 2000, vol. 7.

[26] M. L. Overton, *Numerical computing with IEEE floating point arithmetic*. SIAM, 2001.

[27] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[29] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, "Neural network-based self-tuning pid control for underwater vehicles," *Sensors*, vol. 16, no. 9, p. 1429, 2016.

[30] N. Passalis and A. Tefas, "Neural bag-of-features learning," *Pattern Recognition*, no. 64, pp. 277–294, 2017.