A Real-time Ball Trajectory Follower using Robot Operating System

Himanshi Yadav*, Siddharth Srivastava†, Prerana Mukherjee† and Brejesh Lall†
*National Institute of Technology, Tiruchirappalli, India
† Department of Electrical Engineering, Indian Institute of Technology, Delhi, India
*himanshimj@gmail.com, †{eez127506, eez138300, brejesh}ee.iitd.ac.in

***Abstract*—Modern sports events involve multiple cameras recording the event with very high quality of video. The multiple camera system is complex and generally requires a team of trained men to get the best shot of the ball especially in challenging cases such as line calls in tennis events, corner detection in football events etc. This paper suggests a simple, economical and less time consuming solution using an object tracking algorithm implemented via an Unmanned Aerial Vehicle (UAV). The UAV used in the paper is the Vertical Take-off and Landing (VTOL) Parrot A.R. Drone 2.0 which acts as a moving aerial platform and aims at tracking fast moving ground objects by processing real-time camera feed. Presently, the images from the front camera of the drone are obtained and the moving ground object detected by colour segmentation techniques. The detected object is continuously tracked by centering the image frame. Open Source Computer Vision libraries (OpenCV) are used to process the images obtained from the drone which is controlled by an environment created using Robot Operating System (ROS).**

Keywords- Ball follower, Unmanned Aerial Vehicle, Robot Operating System, PID controller

I. Introduction

Most popular sports events mainly revolve around a fast moving ball and important decisions during these events rely on advanced algorithms to track and detect the exact position of this ball. Hawk-eye is a commonly used technology in sports events like cricket, tennis, Gaelic football, badminton, hurling, and association football where six high performance cameras are utilized to track the ball from different angles. This technique is very popular and is trustfully used as an impartial second opinion during matches. In-spite of the popularity, the above mentioned system is quite complex and requires high-quality expensive cameras and devices to operate [1]. This paper provides an inexpensive solution to the above problem where a Vertical Take-off and Landing (VTOL) Unmanned Aerial Vehicle (UAV) is employed to record and process videos of the sports events in real-time. The UAV used is a relatively inexpensive Parrot A.R. Drone 2.0, which is used to track and follow the fast moving ball throughout the length of the sports event by visual feedback control [2] using relatively low-quality videos.

In [3], authors use a multi-camera system involving nine cameras to track the tennis player's positions after background subtraction and hysteresis-type blob tracking. Feasibility of a multiple camera system in soccer events is investigated in [4] and [5]. These works indicate two major advantages of a multiple camera system i.e. the reduction of occlusion errors and the capability to avoid perspective. Additionally, [5] evaluate the system accuracy and robustness by comparing the estimated ball positions and phases with manual ground-truth data of real soccer sequences. Authors in [1] explore the field of multiple camera usage in baseball but suggests a low-cost method compared to [4], [5] and [3] by using multiple web cameras instead of expensive high-speed cameras with the aid of client-server connections. All these methods suffer from under utilization of resources and therefore the solution proposed in this paper involving only a single aerial camera is more resourceful and thus proves to be drastically cheaper. Recent work similar to ours use moving camera [6][7][8] but require high amounts of processing and thus are not applicable in a scenario where low processing speeds can result in the loss of the current position of the object and a subsequent failure of the tracking task. Therefore this paper focuses on developing a real time ball tracking algorithm which is computationally efficient as well as accurate.

In this work, videos for different sports events are taken from a Parrot A. R. Drone 2.0. These videos have varying amounts of occlusion, ambient light effects, wind effects, video stability etc. The front camera of the drone is employed to take the above mentioned videos. Communication is maintained between a substation computer and the drone via Wi-Fi which ensures fast processing speeds on a remote computer without draining the power bank of the drone. This substation successfully displays processes and also records the videos taken by the drone. OpenCV libraries are utilized for image processing within the ROS framework. A closed loop model is build based on basic proportional integral derivative (PID) control and it operates on the output taken from the processed video.

In the subsequent section, we provide the system description, the proposed methodology is discussed in Section III, the experimental results are provided in Section IV and the conclusion and future work is given in Section V.

II. System Description

A. Mechanical System

The UAV employed for all experiments and related research work was the Parrot A.R. Drone 2.0. It is a lightweight, highly economical and well-equipped drone. There are two cameras that can be used to obtain live feed from the UAV; the front camera is a 720p sensor with 92°diagonal lens that records up to 30fps and the bottom camera is a QVGA sensor with 64°lens that records up to 60fps essentially capable of ground speed measurements. The quadrotor comes with internal in-flight controllers and emergency features making it stable and safe to fly [9]. Due to these aforementioned characteristics, we utilize the real-time videos taken from the front camera of the Parrot A.R. Drone during a stable flight.

B. Software System

The processing is done on a computer running on Ubuntu 14.04 that communicates with the UAV using WiFi.

1) ROS interface: The experimental results were obtained from a system built using the Robot Operating System (ROS) [10] through the Autonomy Lab ardrone_autonomy package. ardrone_autonomy is a ROS driver for Parrot A.R. Drone 2.0 and is based on the official AR-Drone SDK version 2.0.1. The driver takes care of all the low-level communication and can be accessed by commands like takeoff, land, reset and velocity. The controller for the quadrotor motors and the ardrone driver comprise of the three ROS nodes which communicate with each other using simple ROS topics.

2) Image Processing: OpenCV functions are extensively used for carrying out basic image processing operations of object detection and tracking in a separate ROS node.

3) Control System: The output of the processed image is used as the input to the PID controller that provides a new drone velocity as the output.

III. Proposed Approach

In this section we give a detailed overview of the automatic tennis video annotation framework. The workflow of the proposed method is shown in Figure 1. In the following subsections we describe the components in detail.

A. Image retrieval

The front camera of the UAV provides real-time videos and acts as the moving platform. The video is accessed from the ardrone driver by sending *ROS messages* via *ROS Topics* to the drone. The frames in the videos are then processed in the *OpenCV ROS Node* and results are subsequently sent to the *Controller ROS Node*.

B. Image processing

The overall architecture of the steps followed is given in Figure 2.

1) Color Segmentation: Color segmentation is performed with HSV color as it works well with shadows and lighting variations. The hue component depends on the color and varies from object to object whereas the other two components (S,V) depend upon the lighting as well as the surface of the object.

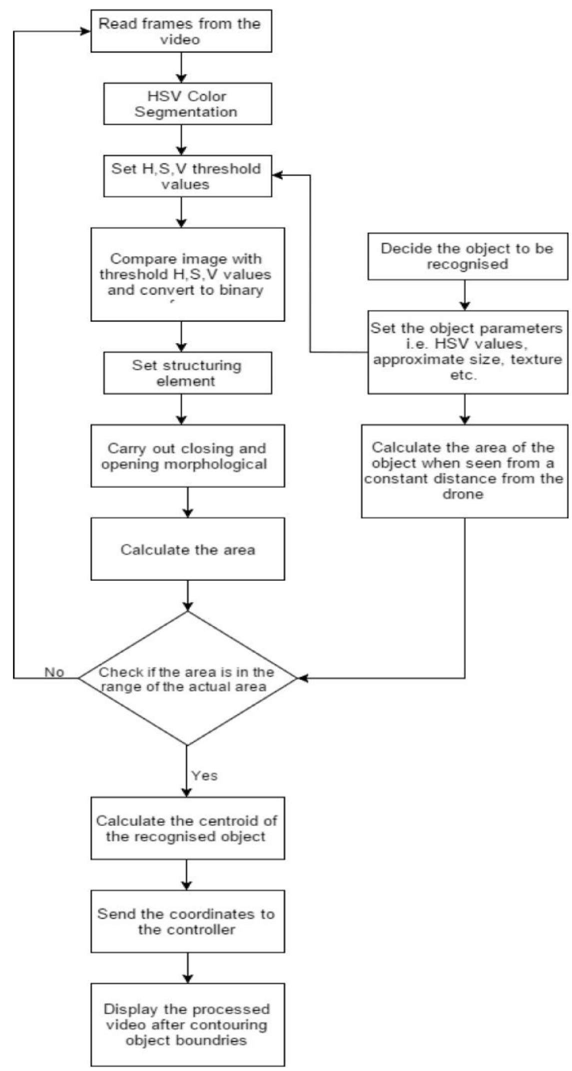

Fig. 2. Steps in Image Processing

2) Morphological Operations: Segmentation from the previous step introduces artifacts in the image in form of disconnected components. These artifacts appear as isolated black and white spots. The isolated black spots in individual thresholded images are removed by carrying out morphological opening. We perform erosion followed by dilation using a (5,5) sized structuring element. The white spots are primarily present inside the main object which are removed by morphological closing which is dilation followed by erosion using the same structuring element.

3) Object Recognition: For object recognition, the locations of the object are noted to create a trail of the moving object from consecutive frames. These are then characterized by the following properties:

1) Centroid
2) Area
3) Orientation
4) Blob Eccentricity
5) Color Composition

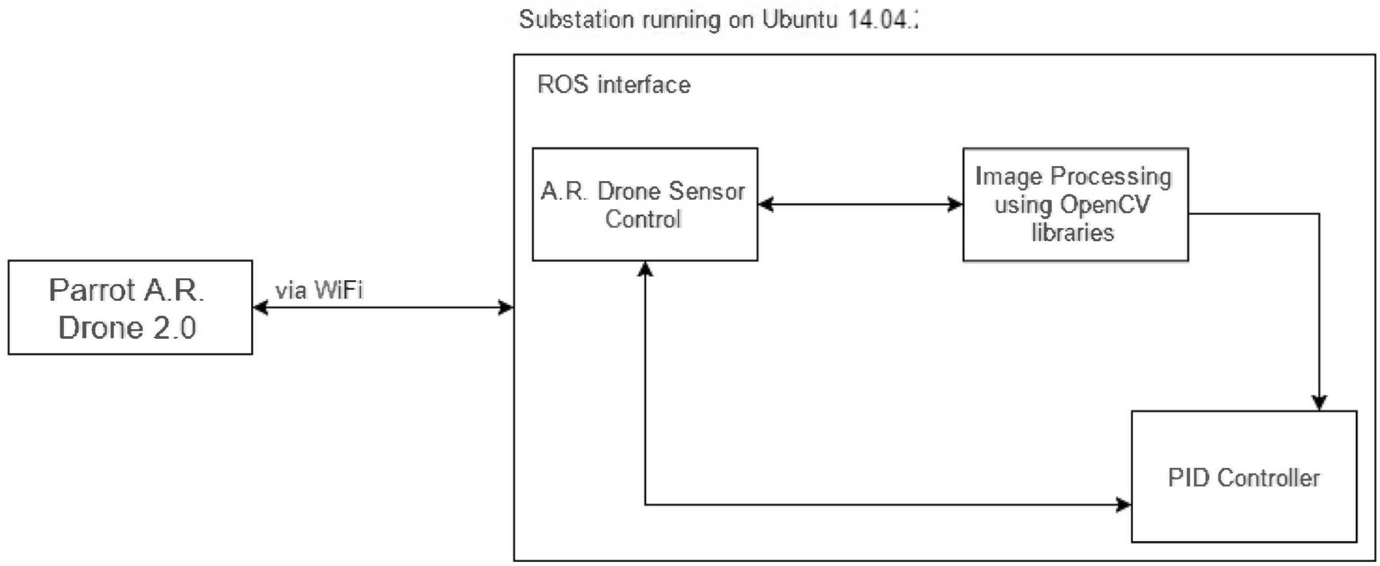

Fig. 1. Overall System Architecture

Image moments (Equations 1 and 2) are then used which are a certain particular weighted average (moment) of the image pixel's intensities. If the area of the recognized objects is less than a threshold then that object is dropped. Using moments the center of the main object that is to be tracked is furnished and sent to the controller node along with other information from the ardrone driver.

$$C_x = \frac{M_{10}}{M_{00}} \tag{1}$$

$$C_x = \frac{M_{01}}{M_{00}} \tag{2}$$

where C_x, C_y are the coordinates of the centroid, M_{01}, M_{10} and M_{00} are the image moments.

4) Object Tracking: The coordinates of the identified object are represented as x_{pos} and y_{pos} and passed onto the control node in ROS. The centre of the the video frame as seen from the front camera of the UAV is pre-defined and therefore the error between x_{fix}, y_{fix} and x_{pos}, y_{pos} respectively is calculated as

$$e_x = x_{pos} - x_{fix} \tag{3}$$

$$e_y = y_{pos} - y_{fix} \tag{4}$$

where e_x and e_y are the errors in x and y directions respectively, x_{pos} and y_{pos} are the coordinates of the cetroid of the object, x_{fix} and y_{fix} are the coordinates of the center of the video frame as depicted in Figure 3.

5) Control System: The Parrot A.R. Drone 2.0 is being very extensively used in research work due to its highly advanced controller and efficient low-level handling of data from numerous sensors. In the proposed system, ardrone_driver controls the various sensors such as the magnetometer, odometer, IMU etc. and publishes the state of the drone using the navigation data variables on the topic ardrone/navdata. The state of the drone is represented by a matrix X in the Euclidean coordinate system having rotational coordinates as below:

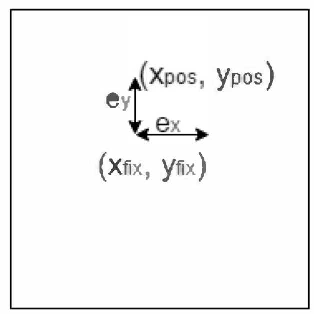

Fig. 3. Figure depicting the coordinates of the center of the video frame

$$X = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

where R is the rotation matrix having three degrees of freedom i.e. θ, ϕ and ψ; t is the translation matrix having three degrees of freedom i.e. X, Y, Z; r_{ij} and t_i are the elements of the matrices R, t respectively. The drone lacks a stationary reference coordinate system and therefore a rotation matrix is used to represent invariance in the motion of the drone.

The following equations depict the rotation matrix of yaw, pitch and roll in terms of the elements of the rotation matrix.

$$R = R_Z(\psi) R_Y(\theta) R_X(\phi) \tag{6}$$

$$\phi = Atan2\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \tag{7}$$

$$\psi = -Atan2\left(\frac{r_{21}}{cos\phi}, \frac{r_{11}}{cos\phi}\right) \tag{8}$$

$$\theta = Atan2\left(\frac{r_{32}}{cos\phi}, \frac{r_{33}}{cos\phi}\right) \tag{9}$$

where ψ represents yaw, θ represents pitch and ϕ represents pitch. The default roll (ϕ), yaw (ψ) and pitch (θ) values are set apriori for each autonomous flight depending upon the environment. The drone is constrained to move only along two axes i.e. forward or backward and sideways. Thus the elevation of the drone is decided before making the flight to maintain the constrained 2-D motion. Each time the object to be recognized moves, the controller produces an output which results in the change in the position and orientation of the drone according to the error in its position so as to centre the object to the frame of the video. A cascaded PID controller is used to directly link the internal PID controller. In the cascaded controller, the low-level PID controller (the internal controller) is cascaded inside the visual feedback controlled PID controller (the external controller). The entire system is modelled as a linear system [1] and controlled using a simple PID controller. The PID equation is arrived at for both the axes i.e. x and y by realizing the K_p, Ki and Kd values for both axes separately as given below,

$$K_p e(t) + K_1 \int e(t)\,dt + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t) \tag{10}$$

where $e = Setpoint - Input$, K_p, K_i and K_d are the positive coefficients for the proportional, integral and derivative terms respectively. The controller converts the error in the position of the drone to a new velocity for the drone in x and y axes which is conveyed to the ardrone_driver using the cmd_velocity topic. Figure 4 gives the block diagram of the PID controller.

IV. Experimental Results and Discussions

A. Ball Detection

The technique discussed in this paper was applied to real-time videos of different sports events (example: tennis, football, basketball) in different environments i.e. indoors and outdoors, with and without ambient light effects, with and without occlusion etc. It was observed that the velocity of the ball, the size of the ball and the constant distance (drone being moved in a constrained plane) between the front camera of the drone and the ball in the field were directly proportional to the ball detection accuracy. Figure 5 gives an example of ball detection. Figure 6 shows an example with change in ambient light conditions. The algorithm however fails when the velocity of the object is very high, when the direction of motion of the ball changes abruptly due to ball-player interaction and in highly occluded frames. While testing under certain unconstrained outdoor environments, we found that if any single tracking algorithm with only one tracker for tracking the motion of the ball is applied; then it proves to be highly inefficient. This occurs due the high speed and unpredictable location of the ball during a sports event. Therefore, further experimental work is needed to make the system more robust. The use of multiple trackers on a particular object and then following the tracker with the least error would ensure that tracking is successful in different cases.

Fig. 5. Figure shows the original and the thresholded images of an object with no occlusion and under indoor environmental conditions.

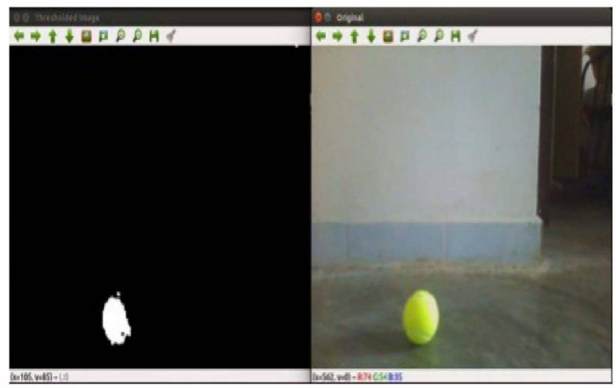

Fig. 6. Figure shows the thresholded image of the same object with no occlusion but with a variation in the ambient light. Due to this the thresholded image does not truly represent the original object. But the algorithm successfully achieves to recognize the object by assuming the position of the current object to be in the vicinity of the original object in the subsequent frames of a video and therefore not losing track of the object due to ambient light effects.

B. Detection accuracy with change in ball velocity

Figure 7 shows that as the velocity of the ball increases the number of frames required to detect the ball also increases. Therefore, the detection accuracy decreases. A solution to this problem is explored in [11] and demands further experimentation.

C. Processing Speed

As can be seen from the Table I time taken to detect any object is very small; this provides sufficient reaction time to the drone and for better real-time performance.

V. Conclusion and Future Scope

In this paper, we have proposed a novel real time and cost-effective approach to record and analyze sports events using a UAV. The ball tracking algorithm and the controller design were proposed. It was found that the proposed algorithm

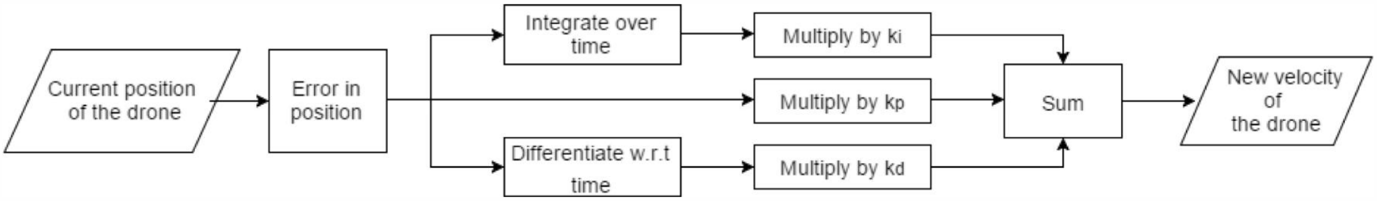

Fig. 4. Block Diagram of the PID controller

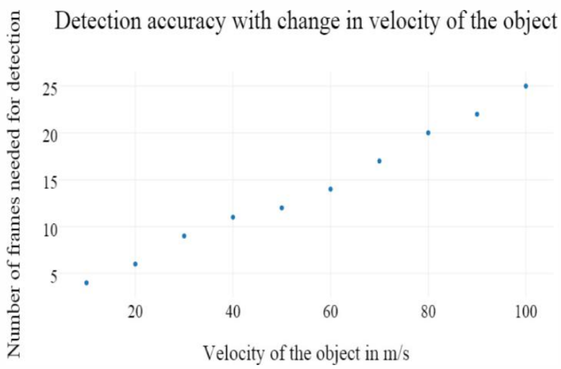

Fig. 7. Figure shows the detection accuracy with change in velocity of the object.

TABLE I
PROCESSING SPEED

Operation	Time taken(in milliseconds)
Image acquisition	1
Colour segmentation	1.5
Ball localisation	1.2
Total	3.7

is faster than the other contemporaries but failed when the velocity of ball was too high. The usage of multiple trackers for the same object was also theoretically explored.

The paper has considered several constrained situations to test the efficiency of the solution being suggested . The solution faces several limitations where the most deterring limitation is the lightweight of the drone which makes it susceptible to any change in the wind velocity and other external environmental conditions. The stability of the video is affected majorly due to the former limitation and both hardware and software stabilization tools are required to make the solution model more efficient. Furthermore, several necessary adjustments are to be made in the mechanical design of the drone to allow more flexible environmental conditions and have a robust system. Despite these the algorithms perform with high degree of precision

The solutions of this paper can theoretically as well as practically be extended to a swarm of drones [12] decentrally achieving the same objective. Also, the quality of the video feed obtained from the drone can be improved by attaching better video capturing instruments with enhanced stabilization techniques at the cost of the weight of the drone and complexity of the system which can make live broadcasting of sports events through autonomously flying vehicles a possibility.

REFERENCES

[1] S. J. You, P. H. Truong, S. H. Ji, S. M. Lee, C. E. Lee, and Y. J. Cho, "A cooperative multi-camera system for tracking a fast moving object," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on.* IEEE, 2014, pp. 141–145.

[2] J. Hwang, Y. Ooi, and S. Ozawa, "A visual feedback control system for tracking and zooming a target," in *Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on.* IEEE, 1992, pp. 740–745.

[3] C. Ó. Conaire, P. Kelly, D. Connaghan, and N. E. Connor, "Tennissense: A platform for extracting semantic information from multi-camera tennis data," in *Digital Signal Processing, 2009 16th International Conference on.* IEEE, 2009, pp. 1–6.

[4] T. D'Orazio, M. Leo, P. Spagnolo, P. L. Mazzeo, N. Mosca, M. Nitti, and A. Distante, "An investigation into the feasibility of real-time soccer offside detection from a multiple camera system," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 12, pp. 1804–1818, 2009.

[5] J. Ren, J. Orwell, G. Jones, M. Xu *et al.*, "Real-time modeling of 3-d soccer ball trajectories from multiple fixed cameras," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 350–362, 2008.

[6] P. Angelov, C. Gude, P. Sadeghi-Tehran, and T. Ivanov, "Artot: Autonomous real-time object detection and tracking by a moving camera," in *Intelligent Systems (IS), 2012 6th IEEE International Conference.* IEEE, 2012, pp. 446–452.

[7] M. M. Kassir and M. Palhang, "A region based camshift tracking with a moving camera," in *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on.* IEEE, 2014, pp. 451–455.

[8] A. W. N. Ibrahim, P. W. Ching, G. G. Seet, W. M. Lau, and W. Czajewski, "Moving objects detection and tracking framework for uav-based surveillance," in *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on.* IEEE, 2010, pp. 456–461.

[9] P.-J. Bristeau, F. Callou, D. Vissiere, N. Petit *et al.*, "The navigation and control technology inside the ar. drone micro uav," in *18th IFAC world congress*, vol. 18, no. 1, 2011, pp. 1477–1484.

[10] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[11] G. Xiang, "Real-time follow-up tracking fast moving object with an active camera," in *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on.* IEEE, 2009, pp. 1–4.

[12] J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, R. Suarez-Fernandez, and P. Campoy, "A system for the design and development of vision-based multi-robot quadrotor swarms," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on.* IEEE, 2014, pp. 640–648.