Advances in Smart Systems Research ISSN 2050-8662
Vol. 6. No. 2 : pp.55-64 : k17is-224

A Method for Performing Efficient Real-Time Object Tracing for Drones

Ioannis Dramouskas[1], Isidoros Perikos[1,2] and Ioannis Hatzilygeroudis[1]

[1] University of Patras, Patras, Greece
{daramousk, perikos, ihatz}@ceid.upatras.gr
[2] Technological Educational Institute of Western Greece, Patras, Greece
iperikos@teiwest.gr

Abstract *Nowadays a big variety of methods and applications are being designed with the aim to assist robotics in autonomously performing tasks. One kind of these robots are the unmanned aerial vehicles or drones, which recently have attracted the interest of the research community. Drones can perform various applications such as reconnaissance, surveillance, search and rescue and infrastructure inspection. A core task in these applications, which is of great importance to be performed accurately, concerns the visual detection, recognition and tracking of specific desired objects. In this work, our aim is to develop a system that makes drones automatically perform surveillance and efficiently detect and track objects using their own camera in real world conditions. The results from the performance evaluation were very encouraging.*

1. Introduction

The flight of unmanned aerial vehicles (UAVs) or drones as they are called today can be operated by various degrees of autonomy either autonomously by onboard computers or under remote control by human operators. Drones are divided into categories based on their wing shape and body structure such as fixed wing, the rotary wing and the multi rotor. Drones are able to hover, take off, land and fly in narrow areas, facts that make them have high maneuverability and most commonly used for scientific researches. Drones can be used for various operations in real world such as aerial photography, search and rescue missions, counting wildlife, delivering medical supplies to inaccessible areas, forest fire detection, surveillance and much more [9][10]. Even in educational procedures, drones can be used by in higher educational institutes and universities in computer vision and robotics courses to assist students in studying planning and computer vision procedures in real world conditions [7]. Most of the UAV's are equipped with single board computers, sensors, actuators, cameras and GPS. As a result they are a powerful tool in the robotics field. Scientific researches that mentioned above focused on algorithmic implementation and hardware usage of quadcopters such us sensors and cameras.

One of the most important operations, that has to be efficiently conducted by drones, concerns the detection and tracking of objects in real world [3]. It is necessary to provide drones with methods to perform automatically and efficiently the detection of specific object [4]. In general these methods intend to recognize objects in images with accuracy and it is a core task in computer vision, necessary in a wide spectrum of applications [6]. Object tracking, similar to object detection is an important field of computer vision and intends to track an object in a sequence of images or frames. The combination of these two is used in several applications such as surveillance in cities, fire inspection in forests and many others.

However, the fast and accurate detection and recognition of desired, specific objects in real world is considered a very challenging and difficult operation. The main difficulty is that drones operate in dynamic environments that are changing continuously due to weather, lighting condition and human interaction and even in many cases by the movement of various objects in the environment. In this work, we present the development of a method that enables drones to efficiently detect and track desired objects in real world conditions. The method analyses and searches image frames provided by the camera of the drone for objects of desired characteristics in terms of color and size and if there is no object in the current view, the drone will rotate and move in order to locate it. Regarding the image analysis of the frames from the camera, the BGR color space image is converted to HSV color space and after that that use the new HSV image is used to filter out all the areas that don’t match to the desirable thresholding depending on the color of the specific object that we search. The resulted image is a binary image where white pixel areas represent the areas that match the color specification. In the finally step of the method, the position of the target object in the camera frame and the angle of the center of it are calculated and based on them, the drone determines its movement in order to keep tracking the object. The examination of the methods performance in real world studies revealed very interesting results.

The rest of the paper is structured as follows. Section presents 2 related works and after that, Section 3 presents our methodology for automatic object detection and tracking in real world conditions. Section 4 presents a case study illustration of our method and finally, Section 5 concludes the paper and provides directions that future work can focus on.

2. Related Work

This section discusses related work on methods and approaches for object detection and tracking. In the literature, there is a huge research interest on the design and the formulation of methodologies for efficient detection and tracing of objects in real word conditions. A relative work about object detection and tracking by UAVs is presented in [1] where authors use Hough Transform, a common method in the computer vision field, to detect certain shapes in the camera's frame such as circles, squares, rectangles, and other. In the work presented in [5], authors use a form of blob detection that distinguishes objects of the drone's camera by color and report quite interesting results. In [2] authors present an attractive way to detect objects in image frames using the uses TLD (Tracking-Learning-Detection) method. The method assumes that in the tracking process the target object's initial position in the camera frame is known, and then the method aims to identify in the next frames the object while it may be moving in the frame. In the detection process, an ensemble classification schema decides whether the object is in the frame or not. An advantage of this method as authors report is its scalability. In the work presented in [8] authors of the paper, implemented a cloud system for multi object detection and identification using Faster-CNN (Convolutional Neural Network), a Deep Learning framework. They implemented an Extended Kalman Filter for localization and a Probabilistic technique for calculating the drone's position without GPS. As the authors report, it is a highly computationally demanding method for GPU which has great results in the detection and identification of multiple objects.

3. Methodology

In this Section, we present the methodology for automatically tracking specific, desired objects. The main steps of the methodology are illustrated in Figure 1.

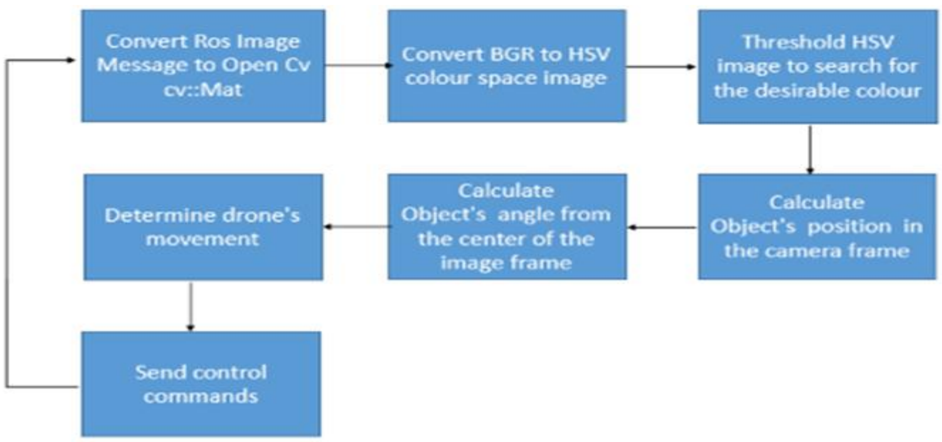

Figure 1. Workflow of the methodology

Initially, an image captured from the drone's camera is converted from the ROS image format to the OpenCV image format so that the image would be available for processing. By default, images taken from the cameras of most drones and also from Bebop 2 that was used in our case study are in RGB (Red-Green-Blue) color space which is their native color space. In general, when searching objects of specific color in images it could be more efficient when the image is represented in the HSV color space. So, the original image is converted from RGB to HSV color space image. Once the image is in HSV space, thresholding process follows next to create a binary image where all the pixels that satisfy the color specifications to be depicted as white and the rest as black. A two dimensional Gaussian filter is used to reduce the noise in the image created by the conversion of the BGR color space image to HSV color space image and the thresholding operation. In addition, erosion and dilation were used in order to filter out the noisy areas of the image and make areas with no interest smaller or even vanish them (erosion) and to make desirable areas bigger and more separable from the other areas of the image (dilation).

Once these steps are completed, the size in pixels of the object is calculated and also the centroid of it, that is the arithmetic mean of all its points, in the object is specified. Using this information, the distance of the object and its angle from the camera of the drone are calculated in order to assist in the determination of the drone's movement so it will be tracking the object. By using OpenCV's function moments it is possible to calculate the centroid of the object. The function computes moments, up to the 3rd order, of a vector shape or a rasterized shape. In order for the algorithm to recognize in the same time more than one objects contour of the object is specified using the algorithm of OpenCV's contour (FindContours) in combination with moments. FindContours algorithm locates the contour of every white area of the binary image, categorized and saves for every area an index which refers to that. By that way every object can be discriminated and analyzed separately from the others in the camera's frame. For each contour, the area which contains it, is calculated and after that, with the help of find max algorithm, the biggest contour is located. For the biggest one, the centre of it is calculated using OpenCV's bounding Rect algorithm, as also its width and height. An example image is given bellow which shows the RGB image, the HSV image and the thresholded image in the Figure 3. When the object is detected, a circle is drawn around it and the calculation of the distance between the object and the drone is estimated and after that, the drone makes its movement decisions as explained in the following section.

3.1. Distance calculation

Due to the fact that Bebop 2 has one single camera which cannot provide information about the depth of the scene, we need to find a way to understand whether the object is moving away or coming closer to the camera. If the target object is detected in the camera frame we are able to calculate its apparent area in pixels. As soon as we calculate the area of the object there are two ways to understand the object's position relative to a previously calculated position. The one way is to compare the object's current size in pixels with the previously calculated size and if the size is bigger it means that the object is moving towards the camera, while if it smaller then it is moving away from the camera. In cases that the size is the same, then the object did not move. Using this way of understanding objects position has disadvantages because we cannot define the distance in order to determine drone's movement and we might lose the object from the image frame. The second method and the one we followed in our method, is based on constantly calculating the distance between the object and the camera. In order to do that, we need to know the object's width in pixels, the object’s real width, the focal length of the camera, and the pixels per/mm on the image sensor. Camera's calibration information is implemented by the ROS package of Bebop 2 named 'bebop autonomy' and is quite important for the constant calibration of the drone. Using the camera calibration data and the size in pixels of the object we can calculate pixels per mm. Finally the distance of the camera and the object can be calculated by the expression (1).

$$d = \frac{object\_real\_width \times focal\_length}{(object\_width\_px / pixels\_px)} \quad (1)$$

Where object_real_width represents the absolute width of the object and focal_length represents the focal value that describes the ability of the optical system to focus light, and is the value used to calculate the magnification of the system. Variable object_width_px represents the width in pixels of the object and variable pixels_px represents the pixels per millimeter on the image sensor.
In order to calculate the angle between the object and the center of the camera frame we have to calculate the degrees of the angle which every pixel expresses in the image using the FoV (Field of View) of the camera and the number of pixels placed in the main diagonal and multiply it with the Euclidean distance of the position in pixels of the object from the center of the image's frame.
The drone can move up or down, backward or forward, right or left and rotate the heading. We choose to avoid the left or right and up or down movement because

the drone has no knowledge of the environment and it may crash so we decide the drone to make only rotation around yaw, and move forward or backward given that the first movement is forbidden to be backwards due to the unknown of the environment behind it. As soon as the distance and the angle are calculated, the drone's movement has to be defined in such a way so the target object is located in the center of the camera's frame. The boundaries where there is no rotation around the heading of the drone are ten degrees for each side. If the angle from the center of the camera's frame is lower than the limit of ten degrees then no rotation follows, other way a command which rotates the drone will be send in order center the object in the camera frame according to the degrees of the angle which was previously calculated. When the target object is centered in the camera's frame we have to specify whether the drone moves slightly towards or backwards according to the distance we calculated. If the distance is between two thresholds set that are 35 to 75cm, the drone will hover. If the distance is bigger than 75 cm it will move forwards until the distance becomes smaller than 75 cm and in the range of the aforementioned thresholds. In cases that the distance becomes smaller than 35 cm, the drone's will slowly move backwards until the distance becomes greater than 35cm. It has to be mentioned that in every circle of calculations we prioritize firstly the rotation to bring the object to the center of the camera frame so in the next circle of calculations it will be centered in order to make the drone moves fowards or backwards according to the process we analyzed above.

In order to implement the method for detecting and tracking objects we used ROS and OpenCV. The "Robot Operating System" (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. We also used the ROS package for Bebop 2 named bebop_autonomy which was the basic tool in order to create the algorithm for the autonomous navigation. Furthermore, in order to analyze camera frames and detect objects in them, we utilized the commonly used library OpenCV. OpenCV (Open Source Computer Vision Library)' is released under a BSD license and hence it's free for both academic and commercial use. The high level programming language we used is C++ because of the compatibility it provides both to ROS and Open CV. As a drone, we chose Parrot's Bebop 2 drone which is depicted in Figure 2. Its equipment includes a front camera at 1080p, stream framerate of 30 frames per second, 3 axis accelerometer for detecting motion, 3 axis gyroscope for detecting attitude and 3 axis magnetometer for detecting heading. In addition, a downward facing ultrasound unit supplies information for altitude when the drone is within 8 meters of the ground while a barometer is used for higher flight, and it is capable to make flights about twenty minutes duration. In addition we used a computer capable of processing the video stream, determine navigational adjustments and send the commands to the drone in real time.

Figure 2. Bebop 2

3.2. Case Study illustration

As an object of interest to be detected by the drone we used a small ball of certain size and color. When this object of interest was placed in a person's hand and the person moved it to different position the drone would be able to track it. In order to make the drone visually follow the target object, the algorithm must be able to distinguish that object from the environment. We achieve that by using color, a method of a big variety of choices by the methods provided from the computer vision field. The algorithm searches every image for a green color object. In this procedure we capture a video stream through the on board front camera of the Bebop 2 and send the images using Wi Fi connection to a laptop for processing. Most of the experiments were conducted main in indoors environments. In Figures 3 and 4, print screens of the moment the object is detected in different light conditions of indoors environments are illustrated.

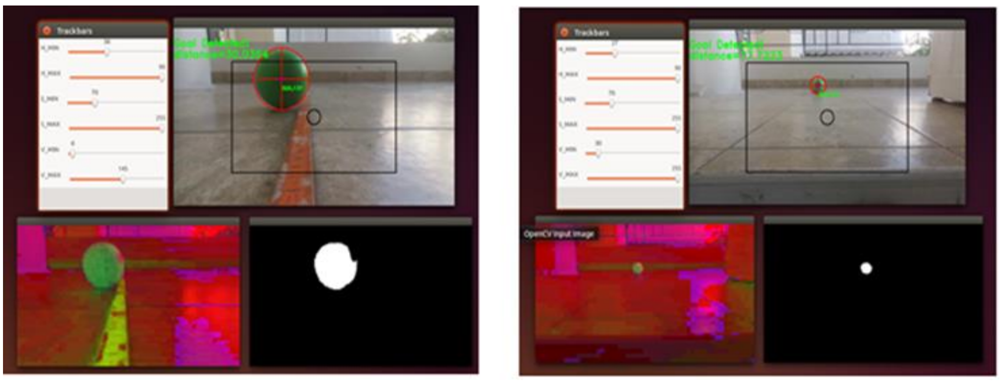

Figure 3.Trackbars, original image, hsv image, threshold image. Target object is detected and the distance is calculated.

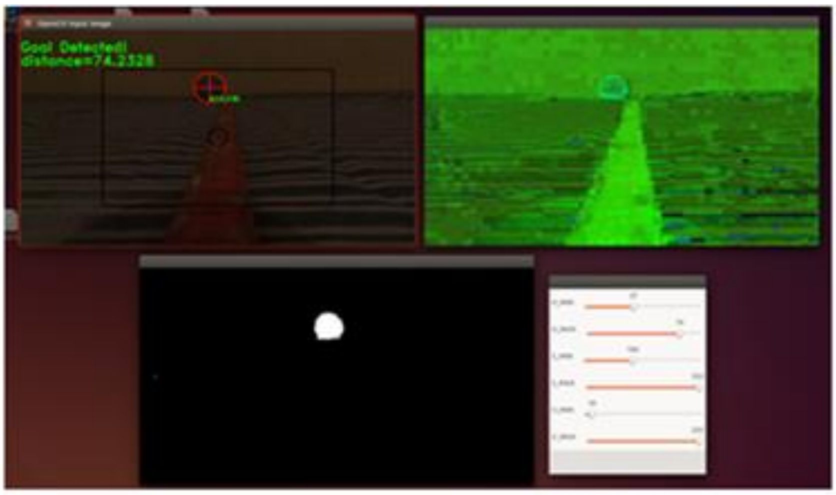

Figure 4.Trackbars, original image, hsv image, threshold image. Target object is detected and the distance is calculated with different light.

Some challenging issues that were raised, were related to lighting conditions of the scenes. So, we have to make sure that the terms of search in the HSV image can be redefined dynamically. To do so, we used trackbars techniques that give our method a dynamic behavior and great adaptation to real world conditions by dynamically changing low and upper values for Hue, Saturation and Value. In this line, Hue is expressed as a number from 0 to 360 degrees representing hues of red (which start at 0), yellow (starting at 60), green (starting at 120), cyan (starting at 180), blue (starting at 240) and magenta (starting at 300), Saturation is the amount of gray from zero percent to 100 percent in the color and Value (or brightness) works in conjunction with saturation and describes the brightness or intensity of the color from zero percent to 100 percent. In that way we are able to detect the desired object regardless of the lighting of the scene in almost every study we conducted.

4. Conclusions and Future Work

In this work, we present a method for efficiently performing detection and tracking of specific object in real world conditions. The method captures consequently frames from the camera in real time and each frame is converted from BGR color space to HSV color space. Then the new HSV image is used to filter out the areas which don't match to the desirable thresholding depending on the color of the desired object that the drone searches. The resulting image is a binary image

where white pixel areas represent the areas that match the color specification. After that, the position of the target object from the drone and the angle of the center of it are calculated and the drone determines its movement to keep tracking the object.
There are various directions that future work will focus on. An interesting aspect to examine concerns the integration of deep learning techniques in order to categorize objects detected and also in the movement description processes. Another direction for future work concerns the design of a system to perform localization by using particle filtering methods in order to assist the drone to localize its position in the dynamic environment. Finally we are interested in developing an optical flow method in order to compute depth estimation dynamically without having to know the objects size in advance.

References

[1] Arifin, F., Daniel, R. A., & Widiyanto, D. (2014). Autonomous detection and tracking of an object autonomously using Ar. drone quadcopter. Jurnal Ilmu Komputer dan Informasi, 7(1), 11-17.
[2] Barták, R., & Vykovský, A. (2015, October). Any Object Tracking and Following by a Flying Drone. In Artificial Intelligence (MICAI), 2015 Fourteenth Mexican International Conference on (pp. 35-41). IEEE.
[3] Boudjit, K., & Larbes, C. (2015, July). Detection and implementation autonomous target tracking with a Quadrotor AR. Drone. In Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on (Vol. 2, pp. 223-230). IEEE.
[4] Chakrabarty, A., Morris, R., Bouyssounouse, X., & Hunt, R. (2016, June). Autonomous indoor object tracking with the Parrot AR. Drone. In Unmanned Aircraft Systems (ICUAS), 2016 International Conference on (pp. 25-30). IEEE.
[5] Chapman, R. F., & Mathias, H. D. (2016). Computer Vision Based Object Detection and Tracking in Micro Aerial Vehicles. Papers & Publications: Interdisciplinary Journal of Undergraduate Research, 5(1), 7.
[6] Flammini, F., Naddei, R., Pragliola, C., & Smarra, G. (2016, October). Towards Automated Drone Surveillance in Railways: State-of-the-Art and Future Directions. In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 336-348). Springer International Publishing.
[7] Krajník, T., Vonásek, V., Fišer, D., & Faigl, J. (2011). AR-drone as a platform for robotic research and education. Research and Education in Robotics-EUROBOT 2011, 172-186.

[8] Lee, J., Wang, J., Crandall, D., Šabanovic, S., & Fox, G. (2017, April). Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles. In Robotic Computing (IRC), IEEE International Conference on (pp. 36-43). IEEE.

[9] Mohammed, C., Taek, B., Khelifa, B., Abdellah, A., Sofiane, A., & Nadia, O. (2016). Unmanned Aerial Vehicles For Objects Detection and Tracking, A Survey.

[10] Petrides, P., Kyrkou, C., Kolios, P., Theocharides, T., & Panayiotou, C. (2017, June). Towards a holistic performance evaluation framework for drone-based object detection. In Unmanned Aircraft Systems (ICUAS), 2017 International Conference on (pp. 1785-1793). IEEE.