Control a Drone Using Hand Movement in ROS Based on Single Shot Detector Approach

Hamed Ghasemi[1], Amin Mirfakhar[1], Mehdi Tale Masouleh[1] & Ahmad Kalhor[1]

[1]Human and Robot Interaction Laboratory, University of Tehran, Tehran, Iran
Email addresses: hamed.ghasemi@ut.ac.ir, mirfakhar_amin@mecheng.iust.ac.ir, m.t.masouleh@ut.ac.ir & akalhor@ut.ac.ir

***Abstract*—These days, drones become one of the most popular robots and have lots of application in delivering, transportation, construction, agriculture, data recording. Nevertheless, the increment in using drones, the rate of world aviation accidents increased too, and most of them were because of hardness to keep the drone in wanted location, and also superficial acknowledgement about multi rotor's structures. In this paper, a new method is introduced for controlling a drone rather than controlling the drone by classical approach via radio control. In this new method, a deep learning algorithm and machine vision applied to detect an object and use its position to order the drone. In the performed experimental study, which was done in Human and Robot Interaction Laboratory, a human hand was used as an object (to simplify the problem) which its movement is made equivalent to the command of a radio controller. In order to accurate the result of hand detection the so-called Single Shot Detector is used which has been trained by using the Ego Hand dataset. The latter leads to a precise result under different conditions like viewpoint variation, background clutter, illumination and so on. Furthermore, a stereo camera vision set up used instead of using Kinect or leap motion devices to evaluate the depth of hand which enable to control the drone in a three-dimensional coordinate system. Also, the proposed algorithm are implemented in the Robot Operation System(ROS) and Gazebo were applied to get these positions and simulate the drone. The drone, which is based on px4 as a flight controller- in the Gazebo environment- receives its position from Mavros node which connected the drone to the ROS and could move continuously along x, y, and z axes. According to the results, it can be inferred that the proposed method was much easier and had more functionality instead of previous methods reported in the literature.**

***Index Terms*—drone, Single Shot Detector, ROS, Gazebo**

I. Introduction

Nowadays, Human and Robot Interaction (HRI) has stimulated the interest of many researchers in different filed, Engineering science, social science and even psychology, which has been known as a multidisciplinary concepts. Robotic devices can be classified into different types and drones are the one which are known to have a challenging control problem. Recently, due to advent in both software and harware of such a robotic device, there has been a widespread of applications for drone in different fields, namely, delivering postal boxes. The pioneer of the foregoing application is Amazon for delivering postal boxes for nearby places which is still in progress. Control of this type of robots is a challenging problem. by default people uses joystick and mobile to control them. but it is very interesting to control them high level. As it was first developed, and from a mechanical standpoint, the desired movement of the drone is transfered by means of a Joystick which can be regarded as the first solution passing through the mind. However, by the advent of vision technologies an alternative has been proposed for controlling drone which eases this task, i.e., control a drone using hand movement and gesture. The latter can be regarded as an representative example of HRI.

Most of the paper propounded in the literature regarding this issue are a synergy of machine vision, neural network and deep learning approaches. From the advent made in the recent years in deep learning approach, more emphasizes has been focused on this approach. In this paper, a new methodology has been proposed for the foregoing problems which is based on the so-called Single Shot Detector method. In the previous work different method proposed. In [1], Yangguang Yu et al uses color and simple image processing technique to detect hand and operator. In [2], Sarkar et al. uses a hand gesture sensor called leap motion in order to recognize hand gesture. In [3], Nagi et al. combined the information of head and face movement in order to commands to the drone. In this paper, faces are detected using Haar-like features. In [4], Lee et al. focuses in Deep learning technique to find the gesture of hand. In the foregoing study, 3 deep neural network is used to find the hand situation. In [5], Natarajan et al. introduces a open source library for human drone interaction based on hand gesture. in which a classic machine learning algorithm to end of detecting five hand gesture. In [6], Tsetserukou et al. uses projecting image on the ground and foot gesture to control a drone. In [7], Costante et al. recored a video and considered only user personalizing instead of real time gesture recognition. In [8] kinect sensor is used in order to recognize the skeleton of human and then control the under study drone. In [9] Zhao et al. introduced a web-based interface to control a the drone using hand gesture. in which hand movement are detected using leap motion and send command were using a server for controlling a drone. In [10], different approaches have been proposed to control a drone, for instance voice and hand recognition. For the purposes of this paper, the whole procedure is implemented in ROS. ROS is a flexible framework, support high-end sensor which has an active community. Due to the latter reason, most of well known robotics products has been released based on ROS framework, such as ABB, KUKA and etc [11].

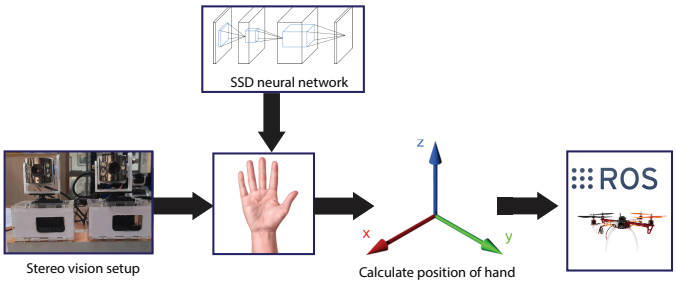


Fig. 1: General overview of the approach used on this paper.

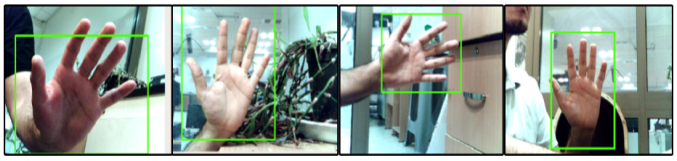


Fig. 2: Detected hand by using SSD.

The main contribution of this paper is two folds: 1) calculating position of hand using SSD and stereo vision and 2) control a drone in ROS using hand movement which is calculated in the previous part.

Figure 1 depicts a general insight about the whole concept used in this paper. Section 2 describes the hand position calculation by means of SSD and upon combining this information with depth image which is obtained from stereo system the position of drone is computed , In the next section the simulation results are demonstrated .section 4 Conclusion.

II. APPROACH

As aforementioned, the proposed approach falls into two part: 1) detecting hand and 2) Using ROS to control the drone. In what follows, first the first part, detecting the hand movement, is explained.

In order to calculate the X, Y and Z of hand two methods can be used:

1) train SSD network to detect hand and calculate coordinates of the hand.
2) uses stereo vision to extract depth image and calculate depth of hand.

A. Train the Single Shot Detector Algorithm

In order to detect the hand movement, deep learning approach is adopted. This technique is very accurate rather than other method such as using color and basic image processing technique which has been used in the previous work [1].
Several deep neural network has been proposed in the literature such as Faster R-CNN [12], yolov3 [13], SSD [14] , ... In this paper, the hand detection should be performed in a real-time manner. The latter is of paramount importance since the drone performs fast movement, consequently requires a fast neural network. i.e., SSD, for hand movement detection.
Single shot detector omits the need of two step detection, which was the common idea behind RCNN, Fast RCNN and Faster RCNN. It uses small convolution kernel usually 3×3 and assigns detection responsibility to it. Not only fast it is, but in addition is more or same accurate as state-of-the-art networks. SSD can be trained end-to-end and it uses only one shot of the picture to detect objects. Inside its structure it uses somewhat feature pyramid which helps detecting small objects.
As a preliminary step in neural network problems, one should collect an appropriate dataset of the under study problem. In the case of this paper, the dataset contains different movement of human hand. For hand detection various dataset such as Oxford Hands dataset [15], EgoHands [16], UPM Hand Gesture Database [17] and etc. In this work, the EgoHands dataset which contains 48 video where each of them have a of 720x1280 size and 90 seconds length.

This dataset has several useful feature as follows::

- images collected from different environment which is useful for generalization problem
- image has a high quality and pixel level segmentation for hand
- unconstrained hand pose

Upon training SSD, for complex situations the network can detect perfectly the corresponding hand movement. Comparing the obtained results from those reported in the literature [ref], it can be concluded that using SSD for hand movement improve considerably the performance and accuracy. It should be noted that most of the studies conducted in this issue is based on simple environment in which no complexity has been involved in the given image to the applied network. Figure 2 illustrates the practical results obtained from SSD for complex environment in which the background of the photo is overloaded. In the left side of this figure it can be seen that the detected hand has an inappropriate situation in light and shape, but the trained network can detect the hand.

B. Stereo Vision for Calculating the Depth

In machine vision, calculating the depth of an image is a challenging problem which requires a stereo vision camera. As shown in Fig. 4, for the purpose of this paper, a stereo vision setup is built. in the below figure we shows the device that we use it for calculate Depth

For extracting the depth of a image the following steps are carried out::

- capture image:
  As the first step, 30 image are captured for both right and left camera. This step is binding for camera calibration. As depicted in Fig. 3., a 7×9 chess board is used for capturing this image.
- Single camera calibration:
  As each camera has some limitations, such as distortion, a calibration is primordial in order to obtain an appropriate image for machine vision purposes. Every camera has radial and Tangential distortion. Radial distortion gener-

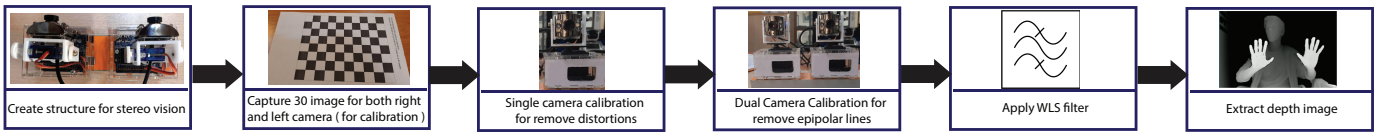


Fig. 3: Flowchart for camera calibration.

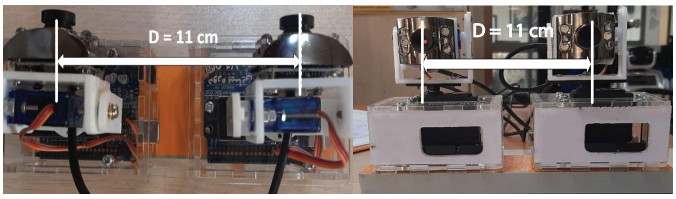


Fig. 4: Stereo vision setup for depth calculation.

ated because of lens shape which can be formulated as follows: [18]

$$x_{\text{distorted}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1)$$

$$y_{\text{distorted}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2)$$

where as x and x are undistored pixel and $x_{\text{distorted}}$ and $y_{\text{distorted}}$ is distorted pixels. In addition k_1, k_2, k_3 are first three term of taylor series.
Also the tangential distortion can be characterized with two parameter p_1 and p_2 with the following equation [18]:

$$x_{\text{distorted}} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (3)$$

$$y_{\text{distorted}} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \quad (4)$$

Tangential distortion accrues because image plane is not parallel with lenses. After the single camera calibration distortion coefficients $(k_1\ k_2\ p_1\ p_2\ k_3)$ can be calculated.

- Dual camera calibration:
  After the single camera calibration for better accuracy stereo calibration is done. the next step consists in aligning the epipolar lines and eliminate the rotation.

- Disparity map filter: for non sparse disparity map a Weighted Least Squares filter has been applied. [19] Moreover a fast global smoother has been applied. because of its speed rather than traditional Weighted Least Squares. [20])

Figure 3 illustrates the whole procedure for calibrating the built stereo vision setup for the purposes of this paper.

III. Simulation

ROS (Robot Operating System) is a powerful framework which contains many libraries and tools which helps ease the task of developing and controlling robots. Thus this tool is used in order to operate and simulate the under study robot. Also, Drones can be controlled in OFFBOARD mode by a companion computer primarily through a set of MAVROS commands which is a higher level wrapper of MAVLink API which saves a great deal of efforts when controlling drones. With the help of MAVROS one can easily realize many functions such as Takeoff, Land, Position Target, Yaw control etc. MAVROS is a communication node based on MAVLink for ROS that is specially designed for communication between the drone and the companion computer. By using this node, some kinds of multi-rotors in Gazebo environment can be launched and connect to them with using nodes. ROS can be used with PX4 and the Gazebo simulator. It uses the MAVROS MAVLink node to communicate with PX4. The ROS/Gazebo integration with PX4 follows the pattern in the Fig. 6 [21] (this shows the generic PX4 simulation environment). PX4 communicates with the simulator (e.g. Gazebo) to receive sensor data from the simulated world and send motor and actuator values. It communicates with the GCS and an Offboard API (e.g. ROS) to send telemetry from the simulated environment and receive commands.
The so-called Iris robot is used for simulating the multi-rotor and connect its flight controller (pixhawk px4) to the ROS by using a python script. The robot can be easily controlled by python code too and get x, y and z for position and yaw angle for orientation.

The result of tracking is shown in Fig. 5.
The blue curve is the position of robot and the red curve is position of hand that move in x, y and z direction.
we define a parameter named x_{diff} that is the mean of difference between robot position and hand position. x_{diff} and x_{diff} is defined similar.
In our result this parameter has following value:

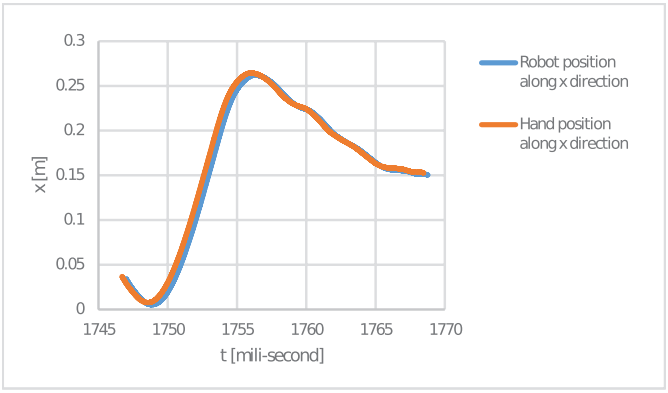
$$x_{\text{diff}} = 0.025m \quad (5)$$

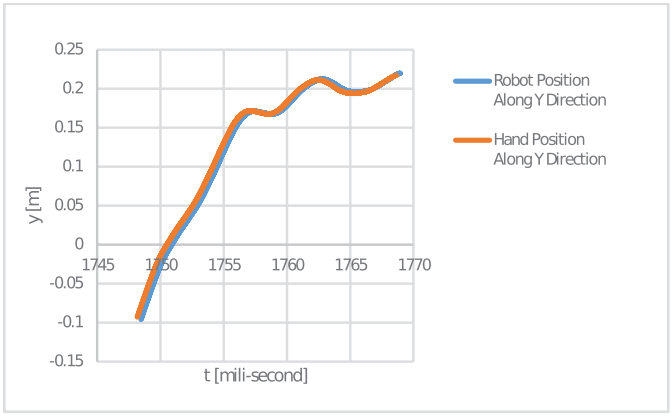
$$y_{\text{diff}} = 0.027m \quad (6)$$

$$x_{\text{diff}} = 0.02m \quad (7)$$

IV. Conclusion

In this paper, a new drone controlling method was proposed which is based on the so-called SSD Deep Neural Network and stereo vision. The deep learning algorithm was trained by Ego Hands dataset to detect the hand and the main novelty of the proposed approach was using this algorithm instead of applying some simple image processing or leap motion techniques which are so sensitive to the point of view and background clutter. Moreover, using stereo camera set up instead of Kinect was the other characteristic of this approach.





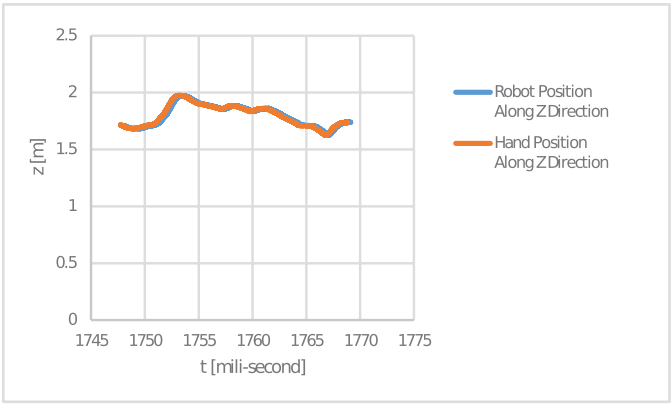


Fig. 5: result of tracking in X, Y and Z.

The propose algorithm was examined and used its results to simulate the drone in ROS and Gazebo environment. Ongoing work consists in increasing the speed of hand detection by applying others deep neural networks and control the real drone and its orientation by hand movements.

References

[1] Y. Yu, X. Wang, Z. Zhong, and Y. Zhang, "Ros-based uav control using hand gesture recognition," in *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 6795–6799, IEEE, 2017.

[2] A. Sarkar, K. A. Patel, R. G. Ram, and G. K. Capoor, "Gesture control of drone using a motion controller," in *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*, pp. 1–5, IEEE, 2016.

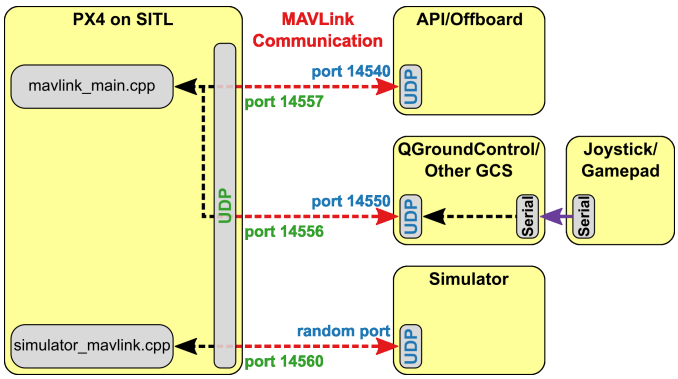


Fig. 6: communication between ROS and drone.

[3] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella, "Human control of uavs using face pose estimates and hand gestures," in *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1–2, IEEE, 2014.

[4] J. Lee, H. Tan, D. Crandall, and S. Šabanović, "Forecasting hand gestures for human-drone interaction," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 167–168, ACM, 2018.

[5] K. Natarajan, T.-H. D. Nguyen, and M. Mete, "Hand gesture controlled drones: An open source library," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, pp. 168–175, IEEE, 2018.

[6] D. Tsetserukou, M. Matrosov, O. Volkova, and E. Tsykunov, "Lightair: Augmented reality system for human-drone interaction using foot gestures and projected image,"

[7] G. Costante, E. Bellocchio, P. Valigi, and E. Ricci, "Personalizing vision-based gestural interfaces for hri with uavs: a transfer learning approach," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3319–3326, IEEE, 2014.

[8] L. Ma and L. L. Cheng, "Studies of ar drone on gesture control," in *2016 3rd International Conference on Materials Engineering, Manufacturing Technology and Control*, Atlantis Press, 2016.

[9] Z. Zhao, H. Luo, G.-H. Song, Z. Chen, Z.-M. Lu, and X. Wu, "Web-based interactive drone control using hand gesture," *Review of Scientific Instruments*, vol. 89, no. 1, p. 014707, 2018.

[10] A. G. Perera, Y. Wei Law, and J. Chahl, "Uav-gesture: a dataset for uav control and gesture recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

[11] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming - Second Edition: Design, Build, and Simulate Complex Robots Using the Robot Operating System*. Packt Publishing, 2nd ed., 2018.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

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

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016.

[15] A. Mittal, A. Zisserman, and P. H. Torr, "Hand detection using multiple proposals.," in *Bmvc*, pp. 1–11, Citeseer, 2011.

[16] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[17] A. I. Maqueda, C. R. del Blanco, F. Jaureguizar, and N. García, "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns," *Computer Vision and Image Understanding*, vol. 141, pp. 126–137, 2015.

[18] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.", 2016.

[19] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.

[20] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[21] “MS Windows NT kernel description.” https://dev.px4.io/v1.9.0/en/simulation/ros$_{i}nterface.html.Accessed$: $2010-09-30$.