Detection and Location Estimation of Object in Unmanned Aerial Vehicle using Single Camera and GPS

Snehil Sanyal
Department of Mechanical Engineering
Defence Institute of Advanced Technology
Pune, India
snehilsanyal@gmail.com

Shashank Bhushan
Center for Artificial Intelligence & Robotics (CAIR)
Defence Research & Development Organization (DRDO)
Bengaluru, India
shabhu18@gmail.com

K Sivayazi
Department of Mechanical Engineering
Defence Institute of Advanced Technology
Pune, India
sivayazi@gmail.com

***Abstract*—It is important for an Unmanned Aerial Vehicle (UAV) to detect any object in its view. This enables the UAV to locate the object with respect to itself and is required for locking and tracking the object. Object detection and location estimation makes the UAV capable of manipulating the environment as well as to follow the target. In the present work, object detection has been carried out using You Only Look Once (YOLO) to detect the object in the image stream of the Robot Operating System (ROS) bag file. The GPS information of the UAV is used to further calculate the GPS coordinates of the object. The images are acquired using a single monocular camera.**

Keywords—computer vision, drone, object detection, depth estimation, monocular camera, ROS, YOLO.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV) are being used for civilian and defence applications like construction, surveillance, disaster management, safety monitoring and much more. They can be driven manually or without any pilot aboard. For these tasks to be done it is important for the UAV to have a clear understanding of its environment. One of the ways to get a clear visualization of the environment is through computer vision. Computer vision techniques enable the UAV to better visualize the environment through images [3]. These images then can be used for detection, localization as well as tracking of objects in the scene [6]. The objects in the scene can be a person, a vehicle or any other moving object. The object then needs to be localized with respect to the UAV so that the drone can manipulate or reach out to the object through waypoints. The location of the localized object in the scene is accomplished through a global coordinate system like GPS or UTM.

Few recent works carried out by other earlier researchers in the area of object detection using UAV and location estimation are reviewed. The combination of inertial navigation system and stereo vision results in a better estimate of object location. In areas where GPS is noisy or unavailable, vision is used as another alternative for the same purpose. Carrillo et al [12] has developed an UAV which is capable of autonomous indoor flight using combination of stereo vision and inertial navigation system. Inertial Measurement Unit (IMU) is used for altitude, rate sensors are used for calculating angular velocities and ultrasonic and pressure sensors are used to calculate altitude at low and high flights respectively. An embedded processor reads the position and altitude signals and computes the control command that achieves tasks demanded by UAV. Stereo Visual Odometry and inertial measurements are fused using KALMAN filter to produce an estimate of vehicle's position, velocity as well as acceleration. The payload is more, hardware cost is high and the algorithm is complex.

Kendall [1] proposed an on-board monocular vision system. The work presents closed loop object tracking control with a low cost on-board monocular vision system and a simple defined target object. An object tracking controller for a quadcopter using an on-board vision system is developed. There are no external localization sensors or GPS. The payload of the UAV increases due to 6 infrared cameras and the algorithm works only for a simple defined object. Therefore, this approach cannot be used for a wide variety of objects.

Zell [11] proposed a Micro Aerial Vehicle (MAV) with 4 cameras which are arranged in 2 stereo configurations, one for Simultaneous Localization And Mapping (SLAM) and one for ground plane detection and tracking. The full 6 DOF pose estimate from each camera pair is fused with inertial measurements in an extended KALMAN filter. For frame tracking Efficient Second order Minimization (ESM) algorithm is used.

Kim and Yow [7] proposed a model for estimation of object location. They used stereo vision using a single camera by taking images of a non-stationary object from 3 different locations increasing the baseline. The object can be of any size and within 11 m distance from the camera. Pedestrian detector algorithm from OpenCV is used.

In view of the above literature, it can be concluded that less work is done in the area of location estimation using single camera [5] because it is impossible to estimate the depth from a single image [2][13][17]. Also, in majority of works the overall accuracy and latency is affected by the choice of object detection algorithm. In the present work, the object is detected from the coming image stream. Using the location of the drone the GPS coordinates of the object is calculated.

II. PROBLEM DEFINITION

Given the home position and GPS location of the drone and the camera parameters the objective is to detect and localize an object in an image with respect to the UAV and

978-1-7281-4997-4/20/$31.00 ©2020 IEEE

calculate the GPS coordinates of the object using Robot Operating System (ROS), Open Computer Vision (OpenCV) and You Only Look Once (YOLO). See Fig. 1 for the overall scenario of the problem.

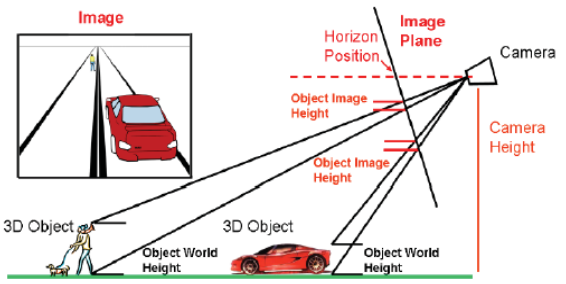

Fig. 1. The mapping of 3D world to 2D image plane using UAV camera [4]. Generally, when images are acquired a physical object is mapped to an image plane. In this process, the depth information is lost because the mapping is from 3D world to 2D image

III. Methodology

Object detection methods which are robust efficient as well as fast are explored. There are two methods for the same: traditional and using deep learning. Traditional methods include contour-based, sliding window, fuzzy-based, graph-based and context-based object detection [10]. Traditional methods depend more on rule-based features and the detection is prone to errors. With the advent of Graphical Processing Units (GPU) and high storage of data, another way is to use deep learning for object detection. Deep learning-based object detection has the advantage of being more robust towards scale change, occlusion, clutter and rotations. It also has the advantage of learning very abstract features [18]. The following networks are used for object detection now a days: R-CNN, Fast R-CNN, Faster R-CNN and YOLO [8][9][14][15][16]. In this work, the object detection is implemented using YOLO.

A. Software environment setup and prerequisites

The work was carried out on a workstation with Ubuntu 16.04 LTS. The software used were ROS Kinetic, OpenCV 3.4 and Python 3.5. The graphics computation was carried out on NVIDIA 1080 TI with Compute Unified Device Architecture (CUDA) toolkit 9.0 and CUDA Deep Neural Network Library (cuDNN) 7.5. The drone was developed at CAIR, DRDO, Bengaluru. The drone has inbuilt sensors including a GPS module and a monocular camera (Leopard Imaging IMX274). The sensors and camera are calibrated. The camera calibration matrix is stored as an YAML file in the ROS workspace. The communication is done using Micro Aerial Vehicle link extendable communication node for ROS (MAVROS).

B. Camera Calibration

Before using the camera in the experiment, it should be calibrated. Camera calibration is done to calculate the camera parameters. The camera model used in this work is the pinhole model. There are two types of camera parameters, extrinsic and intrinsic. The extrinsic parameters map the 3D world coordinates to the 3D camera coordinates. This is given by a rotation and a translation of the camera in the 3D world. These messages are available as the pose of the UAV, since the camera is fixed to the UAV using a gimbal arrangement, the pose of the camera can be calculated using the pose of UAV. Another important set of parameters are the intrinsic parameters. These parameters map the 3D camera coordinates to the 2D image coordinates. This is given by a 3X3 intrinsic camera matrix. The matrix is of the form:

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

where the parameters f_x and f_y are the focal lengths in pixels, c_x and c_y are the image center coordinates or principle point. For calibration of the monocular camera in the UAV, camera_calibration package is used. The camera is calibrated using an 8X6 checkerboard pattern and the script cameracalibrator.py.

C. YOLO for object detection

The drone captures the image stream and publishes the images on a ROS topic. This topic is subscribed by a darknet_ros node. Darknet_ros is a package which enables the darknet framework to run within ROS. The node then publishes a topic which contains an image with the bounding box. This bounding box is created by YOLO with a class name and the confidence score of detection of the class, for example a person with 40% confidence score. The messages that are generated in this process are the bounding box coordinates of two opposite corners, detected image, and a found object message. The following is the result of applying YOLO to an image (see Fig. 2).

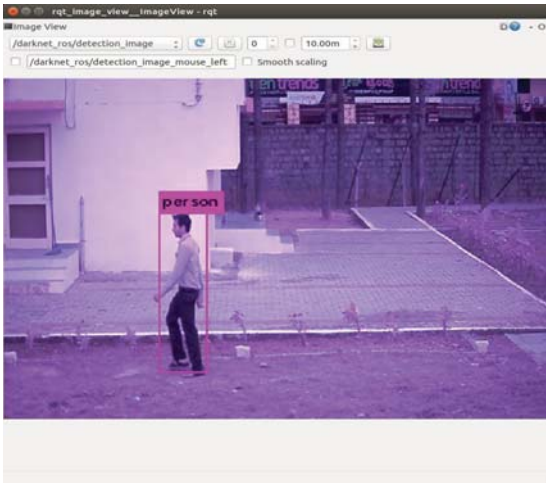

Fig. 2. Detected object and class using YOLO

D. Configuring the network

To remove the misclassifications, the object detection system is configured by reducing the number of classes that YOLO is adjusted to detect. To further increase the accuracy, the threshold confidence score is also increased to 0.5. If the class confidence score is greater than this value, only then the bounding box will be detected.

E. Object depth estimation

It is assumed that the earth is flat, and the bounding boxes are accurate so that the foot of the object in the real world as well as the foot coordinates calculated from the bounding box correspond to the same point. The bounding box coordinates (x_{min} , y_{min}) and (x_{max} , y_{max}) are used to calculate the foot coordinates of the object using (1) and (2):

$$x_f = 0.5(x_{min} + x_{max}) \quad (1)$$

$$y_f = y_{max} \quad (2)$$

These coordinates are the 2D image coordinates of the foot of the object. These pixel coordinates are converted into 2D normalized camera coordinates (x_n , y_n) using the camera parameters f_x , f_y , c_x , c_y . These conversions are given by (3) and (4):

$$x_n = \frac{x_f - c_x}{f_x} \quad (3)$$

$$y_n = \frac{y_f - c_y}{f_y} \quad (4)$$

The depth of the principle point in an image is then calculated using the height of drone and the angle of camera which in this case is $\alpha = 18^o$. The principle point of an image is approximately the image center and this point must correspond to a point on the ground in 3D world (see Fig. 3 and Fig. 4). Using the corresponding depth of the principle point as a reference, the depth of the foot coordinates is calculated. Always the foot coordinates of the object are considered instead of the head or centroid because foot coordinates correspond to a point on the earth which ensures that the altitude information of the foot are not needed. The depth of the principle point is given by (5):

$$H \sec \alpha = Z_p \quad (5)$$

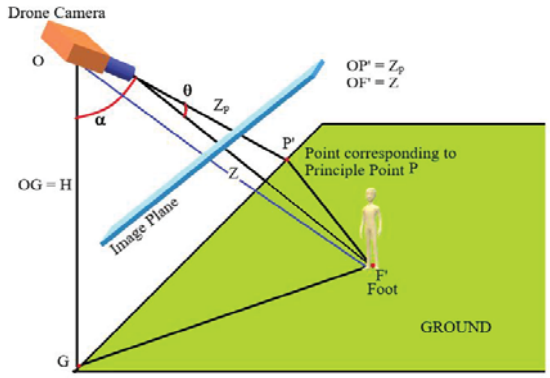

Fig. 3. 3D-view of the problem with the ground and image planes

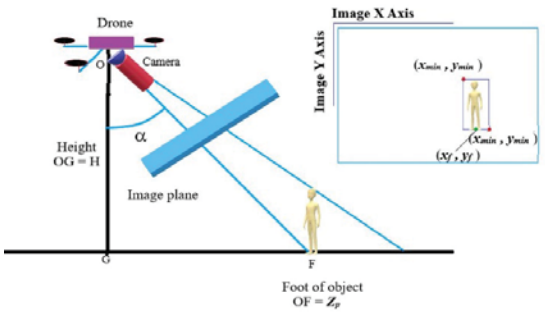

Fig. 4. Depth of the principle point and the image (side view)

Now, the angle subtended by the foot coordinates in image plane with respect to the optical center is calculated. This angle gives the idea of how far the object foot location is with respect to the principle point. The angle can be calculated if the distances OP (focal length) and PF are known. These distances are calculated using Euclidean distance formula (in pixels), whereas the focal length is in mm. Therefore, the distance in pixels are converted to the real-world length units i.e. mm (see Fig. 5 below).

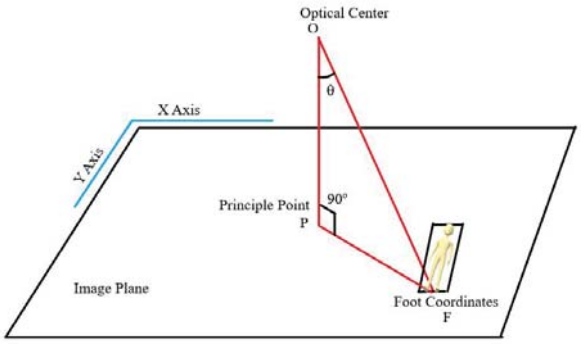

Fig. 5. The distances OP and PF in the image plane

The distance in pixels can be converted in mm using the horizontal and vertical field of views of the camera (see Fig. 6). These parameters are defined using (6) and (7):

$$\tan \angle BOP = \frac{BP}{OP} \quad (6)$$

$$\tan \angle AOP = \frac{AP}{OP} \quad (7)$$

In the above figure, P is the principle point of the image, F is the foot coordinates of the object, OP is the focal length of the camera. OP is perpendicular to the image plane. The angles are $\angle AOP = \frac{VFOV}{2}$, $\angle BOP = \frac{HFOV}{2}$, $\angle OPF = 90^o$. The distances AP and BP are 240 and 320 pixels respectively. To scale the distance, scaling factors in each direction are used. The number of pixels in the horizontal and vertical directions are given respectively by (8) and (9):

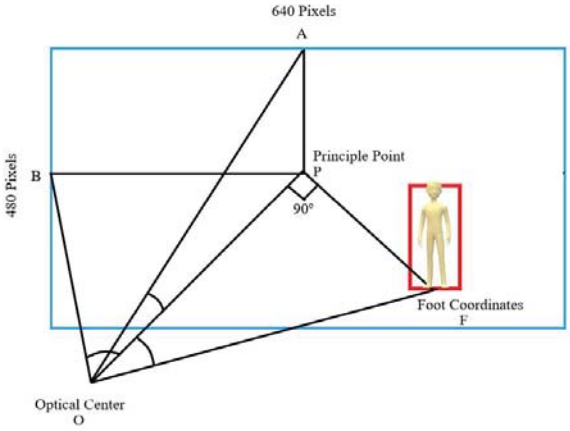

Fig. 6. Horizontal and vertical field of views in the image plane

$$d_x = 1000 * f * \tan \angle BOP \quad (8)$$

$$d_y = 1000 * f * \tan \angle AOP \quad (9)$$

where *f* is in m. The scaling factors in pixels per mm are given by (10) and (11):

$$s_x = \frac{c_x}{d_x} \tag{10}$$

$$s_y = \frac{c_y}{d_y} \tag{11}$$

The distance PF in mm and the angle subtended by the point F (see Fig. 5) are given by (12) and (13):

$$d^2 = \left(\frac{c_x - x_f}{s_x}\right)^2 + \left(\frac{c_y - y_f}{s_y}\right)^2 \tag{12}$$

$$\theta = \tan^{-1}\left(\frac{d}{f}\right) \tag{13}$$

where both *d* and *f* are in same units i.e. mm. The depth of the foot coordinates, which is the distance between the camera and the foot of the object in 3D world in m (see figure 3), is given by (14):

$$Z_p \sec\theta = Z \tag{14}$$

The camera frame coordinates (x_c, y_c, z_c) are calculated using normalized camera coordinates (see Fig. 7) given in (3) and (4) as following:

$$x_c = x_n * Z = \left(\frac{x_f - c_x}{f_x}\right) * H \sec\alpha \sec\theta \tag{15}$$

$$y_c = y_n * Z = \left(\frac{y_f - c_y}{f_y}\right) * H \sec\alpha \sec\theta \tag{16}$$

$$z_c = Z \tag{17}$$

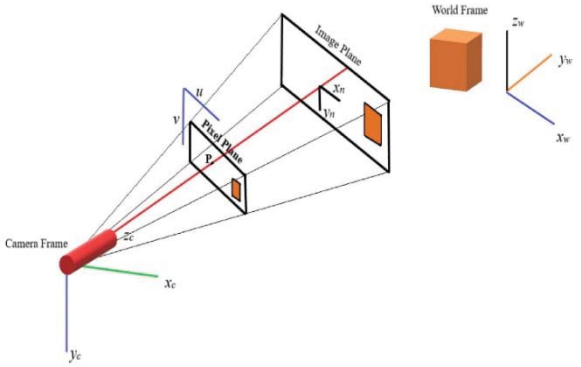

Fig. 7. Camera coordinate frame and the world frame

The 3D camera frame coordinates are converted to body frame coordinates (see Fig. 8), since the camera is tilted at an angle α = 18°. The relationship between the world frame and camera coordinate frame is given by the set of equations (18)-(20) as:

$$x_b = x_c \tag{18}$$

$$y_b = y_c \cos\alpha + z_c \sin\alpha \tag{19}$$

$$z_b = -y_c \sin\alpha + z_c \cos\alpha \tag{20}$$

The body frame coordinates (x_b, y_b, z_b) are then converted to East North Up (ENU) coordinates (see Fig. 9) to have the same reference frame as GPS coordinates. The angle of heading with respect to north is *p*. The object is then projected to the ENU frame to calculate the distance and the bearing from the drone. The relationship between camera coordinate frame and the body frame is given by the set of equations (21)-(23) as:

$$E = z_b \sin p + x_b \cos p \tag{21}$$

$$N = z_b \cos p - x_b \sin p \tag{22}$$

$$U = -y_b \tag{23}$$

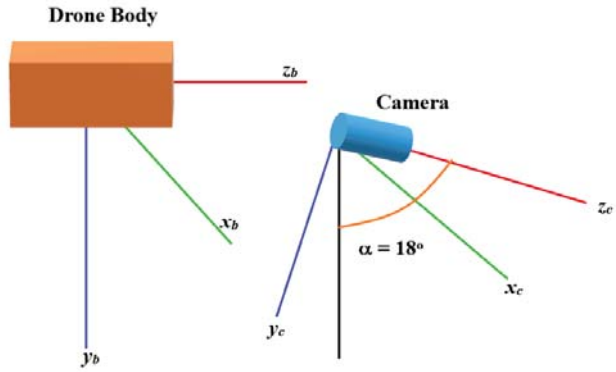

Fig. 8. Camera frame and the body frame

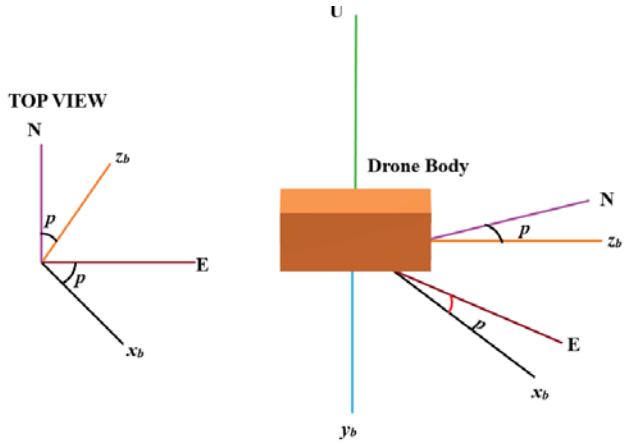

Fig. 9. Body frame and the ENU frame

F. *Location estimation using GPS*

The projection of the object on the ENU frame makes an angle *b* with respect to the north direction in the world frame. This bearing is used to calculate the correction in the GPS coordinates (see Fig. 10). The bearing *b* is given by (24). The correction is applied to the home GPS coordinates of the drone to get the GPS coordinates of the object in 3D world. See Fig. 11 for latitude and longitude representation in earth-centered earth frame (ECEF).

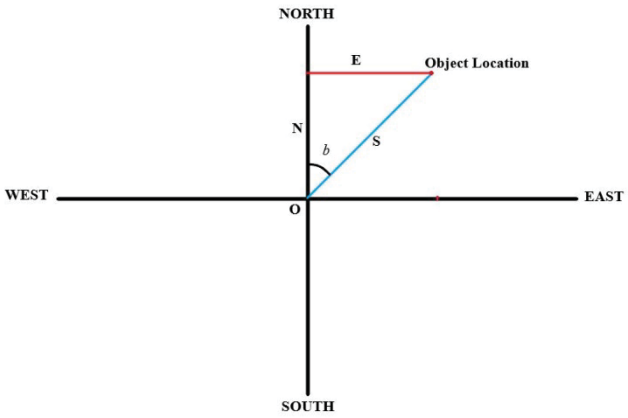

Fig. 10. Bearing of the point with respect to North

$$b = \tan^{-1}\left(\left|\frac{E}{N}\right|\right) \tag{24}$$

The distance of the point from the compass center is calculated using (25):

$$s^2 = E^2 + N^2 \tag{25}$$

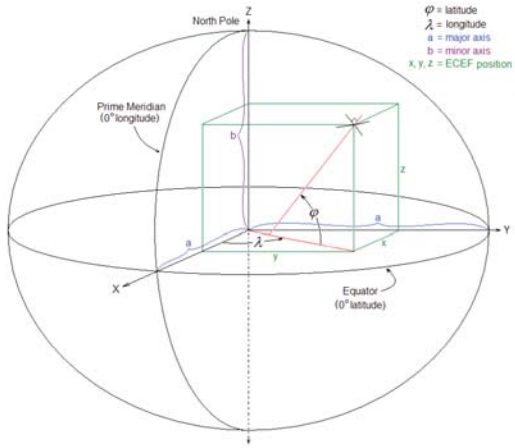

Fig. 11. Latitude and longitude with earth-centered earth frame (ECEF)

The corrections in the X and Y (dX and dY respectively) directions are calculated using (26) and (27):

$$dX = s \sin b \tag{26}$$

$$dY = s \cos b \tag{27}$$

The corrections in the latitude (l_a) and longitude (l_o) are given by the equations (28) and (29):

$$\delta l_o = \frac{dX}{11320 \cos l_a} \tag{28}$$

$$\delta l_a = \frac{dY}{110540} \tag{29}$$

The final latitude (l_{af}) and longitude (l_{of}) are given by (30) and (31):

$$l_{af} = l_a + \delta l_a \tag{30}$$

$$l_{of} = l_o + \delta l_o \tag{31}$$

IV. Results

The object detection has been applied to three ROS bag files which were recorded during the flight of the drone. The objects are successfully identified as persons with a confidence score greater than 0.5. The results are real-time and some of the frames and instances of the overall result are shown below in Fig. 12, Fig. 13 and Fig. 14:

Fig. 12. Image of 2 persons

Fig. 13. Image with detected objects, object name "person" and bounding boxes

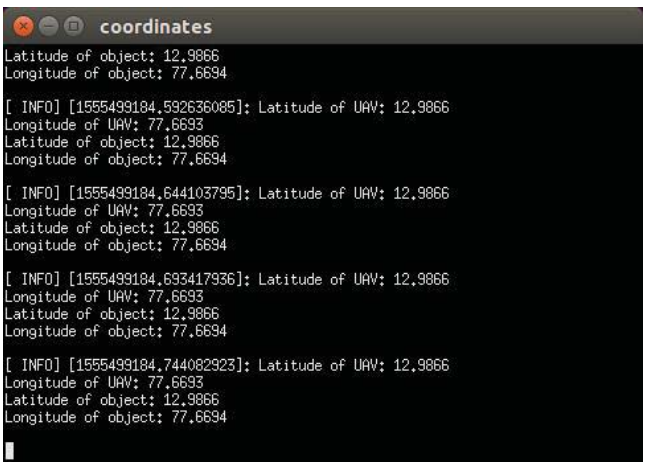

Fig. 14. An instance of the final GPS coordinates of object

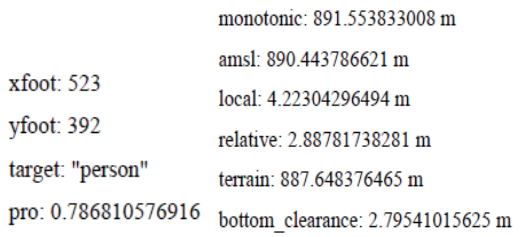

Fig. 15. An instance of the generated messages in ROS

The xfoot, yfoot message is the image coordinate of the foot of the object (x_f, y_f). The foot coordinates of the object is assumed to be equivalent to the midpoint of bottom coordinates of the bounding box.

V. Conclusions

From the results obtained it is evident that the detection of object as well as the estimation of location has been successfully carried out using a single monocular camera and GPS module in the UAV. The work cannot be carried out in a GPS denied environment and the accuracy largely depends on how accurate the GPS signal is received and how many satellite signals are available. The accuracy of the overall model is also subject to the accuracy of YOLO as well as the parameters of UAV like pose and camera matrix. The work assumes flat earth model, the curvature of earth is ignored while estimating the distances but is taken in consideration during calculation of GPS location. The work can be improved by considering different networks for object detection and involving structure from motion algorithm to improve the accuracy of distance estimation.

Acknowledgment

The author acknowledges Center for Artificial Intelligence & Robotics (CAIR), DRDO Bengaluru for giving the opportunity to carrying out the work there. The author also thanks Mr. K. Sivayazi from Defence Institute of Advanced Technology, Pune for his guidance.

References

[1] Alex G. Kendall, Nishaad N. Salvapantula, Karl A. Stol, "On-board object tracking control of a quadcopter with monocular vision", International Conference on Unmanned Aircraft Systems, Orlando, USA, May 2014.

[2] Amlaan Bhoi, "Monocular Depth Estimation: A Survey", arXiv preprint arXiv:1901.09402 [cs.CV], January 2019.

[3] Christoforos Kanellakis, George Nikolakopoulos, "Survey on Computer Vision for UAVs: Current Developments and Trends", Journal of Intelligent & Robotic Systems, vol. 87, issue 1, pp. 141-168, July 2017.

[4] Derek Hoiem, Silvio Savarese, Representations and Techniques for 3D Object Recognition and Scene Interpretation, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2011.

[5] Duc X Tran et al. "Dual PTZ cameras approach for security face detection", IEEE Fifth International Conference on Communications and Electronics, Danang, Vietnam, July 2014.

[6] Fahd Rafi, Saad Khan, Khurram Shafiq, Mubarak Shah, "Autonomous Target Following by Unmanned Aerial Vehicles", Proceedings of SPIE – The International Society for Optical Engineering, vol. 6230, Unmanned Systems Technology VIII, Orlando, USA, May 2006.

[7] Insu Kim, Kin Choong Yow, "Object Location Estimation from a Single Camera", Ninth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Nice, France July 2015.

[8] Joseph Redmon, Ali Farhadi. "YOLOv3: An Incremental Improvement", arXiv preprint arXiv:1804.02767 [cs.CV], April 2018.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, June 2016.

[10] Kartik Umesh Sharma, Nilesh singh V. Thakur. "A review and an approach for object detection in images", International Journal of Computational Vision and Robotics, vol. 7, nos. 1/2, pp. 196-237, January 2017.

[11] Konstantin Schauwecker, Andreas Zell, "On-Board Dual-Stereo-Vision for the Navigation of an Autonomous MAV", Journal of Intelligent & Robotic Systems, vol. 74, issue 1-2, pp. 1-16, April 2014.

[12] Luis Rodolfo García Carrillo, Alejandro Enrique, Dzul López, Rogelio Lozano, Claude Pégard, "Combining Stereo Vision and Inertial Navigation System for a Quad-Rotor UAV", Journal of Intelligent & Robotic Systems, vol. 65, issue 1-4, pp. 373-387, January 2012.

[13] Pablo Revuelta Sanz, Belén Ruiz Mezcua, José M. Sánchez Pena, Depth Estimation – An Introduction, Current Advancements in Stereo Vision, July 2012.

[14] Ross Girshick, "Fast R-CNN", IEEE International Conference on Computer Vision, Santiago, Chile, December 2015.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, June 2014.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, issue 6, June 2017.

[17] Teak-Young Seung, Gi-Chang Kwon, Kwang-Seok Moon, Suk-Hwan Lee, Ki-Ryong Kwon, "An Estimation Method for Location Coordinate of Object in Image Using Single Camera and GPS", Journal of Korea Multimedia Society, vol. 19, issue 2, pp. 112-121, February 2016.

[18] Zhong-Qiu Zhao, Shou-tao Xu, Peng Zheng, Xindong Wu, "Object Detection with Deep Learning: A Review" IEEE Transactions on Neural Networks and Learning Systems, pp. 1-21, January 2019.