Original research papers

UAV Environmental Perception and Autonomous Obstacle Avoidance: A Deep Learning and Depth Camera Combined Solution

Dashuai Wang[a,1], Wei Li[a,*], Xiaoguang Liu[b], Nan Li[c], Chunlong Zhang[a]

[a] *College of Engineering, China Agricultural University, Beijing, 100083, P. R. China*

[b] *Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA, 95616, USA*

[c] *Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, 518172, P. R. China*

ABSTRACT

In agriculture, Unmanned Aerial Vehicles (UAVs) have shown great potential for plant protection. Uncertain obstacles randomly distributed in the unstructured farmland usually pose significant collision risks to flight safety. In order to improve the UAV's intelligence and minimize the obstacle's adverse impacts on operating safety and efficiency, we put forward a comprehensive solution which consists of deep-learning based object detection, image processing, RGB-D information fusion and Task Control System (TCS). Taking full advantages of both deep learning and depth camera, this solution allows the UAV to perceive not only the presence of obstacles, but also their attributes like category, profile and 3D spatial position. Based on the object detection results, collision avoidance strategy generation method and the corresponding calculation

[1] Present address: Guangdong Provincial Key Lab of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, P. R. China

[*] Corresponding authors.

E-mail addresses: ds.wang1@siat.ac.cn (D. Wang), liww@cau.edu.cn (W. Li), lxgliu@ucdavis.edu (X. Liu), linan@cuhk.edu.cn (N. Li), zcl1515@cau.edu.cn (C. Zhang)

approach of optimal collision avoidance flight path are elaborated detailly. A series of experiments are conducted to verify the UAV's environmental perception ability and autonomous obstacle avoidance performance. Results show that the average detection accuracy of CNN model is 75.4% and the mean time cost for processing single image is 53.33ms. Additionally, we find that the prediction accuracy of obstacle's profile and position depends heavily on the relative distance between the object and the depth camera. When the distance is between 4.5m and 8.0m, errors of object's depth data, width and height are -0.53m, -0.26m and -0.24m respectively. Outcomes of simulation flight experiments indicated that the UAV can autonomously determine optimal obstacle avoidance strategy and generate distance-minimized flight path based on the results of RGB-D information fusion. The proposed solution has extensive potential to enhance the UAV's environmental perception and autonomous obstacle avoidance abilities.

KEYWORDS

UAVs, deep learning, depth camera, object detection, environmental perception, obstacle avoidance

1. Introduction

Over the past few years, UAVs, also known as drones, are no longer exclusively associated with military and defense applications, but have been successfully applied in many civilian fields(Floreano et al., 2015), including power-line inspection, rescue aid, crop surveillance (Fernando et al., 2018), crop yield assessment (Feng et al., 2020) and plant protection (Tetila et al., 2020). Plant protection, especially pests and diseases control through spraying pesticide (Ahmad et al., 2020; Liao et al., 2019; Xu et al., 2019), is an important link in the process of agricultural production. Compared with tradition ground-walking plant protection equipment, UAVs have distinct advantages in terms of flexible-terrain-adaptability and high-efficiency (Xue et al., 2016). Currently, with the help of some conventional sensors, reliable control algorithms and obstacle's location information measured in advance, UAVs have already been able to autonomously perform specific tasks along detected or preset flight routes (Basso et al., 2020; Yang et al., 2019). However, there are many unknown obstacles in the unstructured farmland environment, some of them are stationary, others are dynamically moving, which could pose rigorous challenges to the drone's active cognitive ability. So far, it remains a great challenge to endow the UAV with certain environmental perception and obstacle avoidance abilities so that it can automatically generate the optimal collision avoidance strategy and trajectory according to obstacle's specific category, profile and 3D spatial position.

Common challenges in all kinds of applications of UAVs are safety and automation. Many researchers and engineers are committed to eliminating these concerns and making them capable of satisfying the individual requirements on different occasions (Adrian et al., 2020). The top priority for flight safety is that the drones are capable to sense and understand the surrounding environment proactively. The most intuitive way to achieve environmental perception is to obtain as much detailed environmental information as possible. Generally, some common sensors, such as radar, LiDAR, ultrasonic, and infrared rangefinders, have been widely used on UAVs to detect the existence and distance of obstacles (Jongho et al., 2020). However, given inherent limitations like resolution, sensing range and light sensitivity, they can only provide very rough information to UAVs. In addition, monocular cameras are also commonly used on drones. Combined with image processing technology, they can help drones understand the environment in RGB space. But light sensitive and timeconsuming features limit their performance in outdoor applications. Therefore, the lack of knowledge of ambient properties leads to the mismatch between UAV autonomous flight ability and real demand. In the wake of the development of sensors integration and image processing technologies, RGB-D cameras are becoming affordable and applicable for robots to sense the world in higher dimensions. Recently emerged RGB-D cameras like Intel RealSense D435, with visible features of lightweight, high accuracy and light insensitivity, display great potentials to be an effective means to sense flight scenarios. Besides of three channels of RGB information, RGB-D cameras present an extra channel of depth information, which makes it possible to obtain obstacle's color, profile and position features simultaneously. However, how to promptly and effectively extract the most useful information from all features remains a huge challenge. In recent years, some state-of-the-art Convolutional Neural Networks (CNN) and object detection algorithms have been proposed as the prosperity of deep learning (Yann et al., 2018). For example, in terms of classification accuracy and inference speed, YOLO (Redmon et al., 2018) and SSD (Liu et al., 2016) have shown high performance in the field of object detection. Therefore, it would be a wise strategy to extract obstacle's attributes by combining the deep learning algorithms and RGB-D cameras. Many researchers have focused on improving the object detection accuracy by fusing all the information from four channels (Loghmani et al., 2019; Zia et al., 2017). For instance, Single Stream Recurrent Convolution Neural Network (SSRCNN) and Depth Recurrent Convolution Neural Network (DRCNN) to detect and render salient object for RGB-D images were put forward (Liu et al., 2019). Evaluations on four datasets demonstrated that the presented method is excellent in discriminating depth feature and fusing RGB and depth information. Existing studies mainly use depth information to improve classification accuracy. However, in agriculture, there are no reports about the implementation of deep learning and depth cameras on drones to sense the multi-dimensional attributes of obstacles.

As for the automation of UAVs, two important contents are autonomous navigation and obstacle avoidance. Global Positioning System (GPS) usually plays a vital role in navigation systems which guide UAVs with accurate spatial position coordinates. However, GPS signals could be weak or totally lost in some scenarios like urban areas, low altitude flights or indoor operations (Mohta et al., 2018; Perez-Grau et al., 2018). Based on the automatic navigation system, in order to ensure the efficiency and effectiveness, it is necessary to discuss the subject about how to generate and determine the most appropriate strategy to circumvent obstacles with their individual properties in mind. There are various optimization algorithms with different advantages and disadvantages for flight path planning (Shao et al., 2018). However, even with the applications of navigation systems, high-performance sensors and flight path optimization algorithms, it is still challenging for UAVs to reliably perceive the surrounding environment and autonomously navigate between target locations. Furthermore, it becomes more difficult to avoid unknow obstacles with only little or even no prior knowledge of the operating environment.

Aiming at promoting the application of UAVs in the field of plant protection, we develop a novel solution which would be helpful to further ensure operating safety and efficiency by improving the level of intelligence and automation. Contributions of this research can be summarized as follows:

a) A comprehensive solution which consists of deep-learning based object detection, image processing, RGB-D information fusion and task control system is proposed to enhance the UAV's abilities of environmental perception and autonomous collision avoidance.

b) Combining deep learning with depth camera, we put forward a method of RGB-D information fusion. Based on this, the UAV not only can sense the existences of obstacles, but also able to perceive what and where they are.

c) Taking single tree for example, the generation approach of specific obstacle avoidance strategy and the corresponding flight path planning method are elaborated on the basis of the obstacle's attributes.

d) A customized dataset is built to train and evaluate the CNN model with YOLO V3 object detection algorithm.

2. Materials

2.1. Dataset

Since there is no existing open-source dataset containing the specific obstacles distributed in farmland, we establish our own dataset by combining the means of searching online and filming in field. The dataset contains 3,700 samples that can be classified into five categories, i.e., person, tree, building, power line/tower and drone. Each category accounts for the same proportion. For the sake of training CNN model with supervised learning, the categories and bounding boxes of each sample are manually labelled in advance. Because that the inconsistence of image size may cause adverse impact on the model training process, all samples are cropped to the unified resolution of 416×416 before annotating the target objects. Furthermore, the dataset is divided into two parts: the training set and the validation set, which contains 3,000 and 700 samples respectively.

2.2. Workstation

The training and testing processes of the CNN model are implemented on our workstation whose operating system is Ubuntu 16.04 LTS. The major specifications of the workstation are as follows: GPU: NVIDIA GTX1080; CPU: Intel Core i7-8700k; RAM: Corsair 16G; Hard Disk: Samsung SSD 500G. Within the PyCharm developing environment, we build the CNN architecture with TensorFlow computational framework in Python programming language. In addition, the object detection algorithm runs on the GPU which has been configured with CUDA 9.0 parallel programming platform and CuDNN 7.1 accelerating package.

2.3. Simulation environment

A simulation environment, which is composed of Intel RealSense SDK, virtual UAV, ArduPilot, QGroundControl, TCS and customized scripts, is built in the Ubuntu 16.04 LTS operating system. With the help of multiple useful packages, such as ROS, MAVROS, OpenCV, etc., customized scripts are developed for acquiring and optimizing color and depth images, running deep learning algorithms, generating the optimal avoidance strategy, planning flight path and dispatching multi-point flight tasks and obstacle avoidance procedures. In addition, the ground control station named as QGroundControl is employed to observe and record real-time flight parameters and to monitor the executing processes of the flight missions. It is worth noting that the flight control program running on the workstation can be directly transplanted to the flight controller without any modification by right of the hardware compatibility of ArduPiot. This means that the simulation results can effectively represent the actual situation without considering the environmental parameter interference.

3. Systems and methods

The first part of this section presents an overview of our proposed solution. In the subsequent six parts, the methods of object detection, depth data extraction, RGB-D information fusion, obstacle avoidance strategy, flight path planning and autonomous flight control are introduced respectively.

3.1. Overall solution

In order to grant the UAV with certain environment perception and collision avoidance abilities and ensure its flight safety, we propose an overall solution which is shown as Fig. 1.

An Intel RealSense D435 mounted on the UAV is employed to sense the world by simultaneously capturing color and depth images of the flight scene. First, the color image is fed into CNN which has been trained based on our customized dataset to obtain the potential obstacle's classification and bounding box. Then, mapping the results of object detection on the optimized depth image to extract the obstacle's profile and 3D spatial information. By fusing the outcomes of object detection and the extracted depth information, the UAV can determine the optimal avoidance strategy and calculate the distance-minimized obstacle avoidance trajectory according to the obstacle's unique attributes like category, profile and position. Finally, with our novel TCS and customized scripts, the UAV can execute straight-line flight task between multiple task-points while avoiding obstacles autonomously.

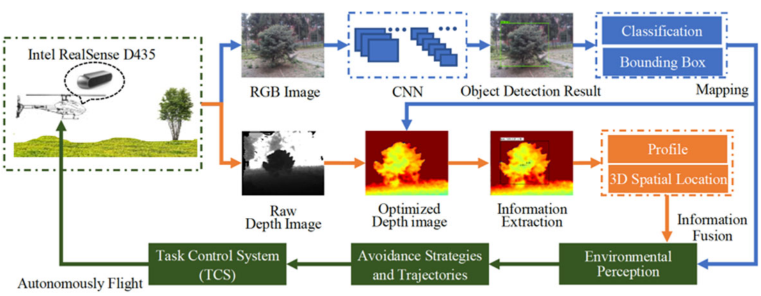


FIGURE 1. An overview of environmental perception and obstacle avoidance solution.

3.2. Object detection

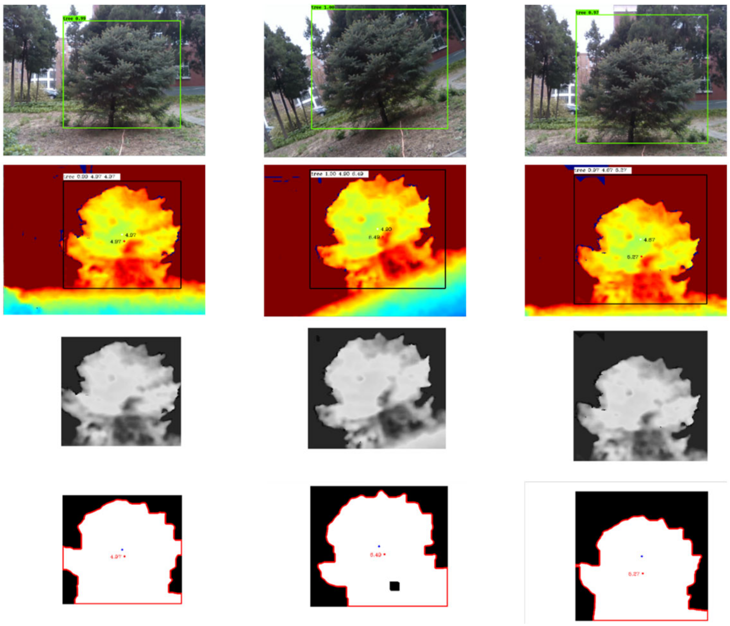
In this study, YOLO V3 (Redmon et al., 2018), one of the state-of-the-art CNN models, is employed to detect the obstacle's category and bounding box. YOLO V3 with the darknet-53 backbone, consists of 75 convolutional layers. And, to non-linearize the model while avoiding overfitting, each one except the last three convolutional layers is followed by Batch Normalization and Leaky_ReLU activation function. By means of up-sampling and concatenation, YOLO V3 can output three feature maps with different scales and the best one would be selected according to the size of potential obstacle for further classification prediction and bounding box regression. Besides, it is especially suitable for occasions with high real-time requirements due to its fast detecting speed and relatively high detecting accuracy.

Generally, the larger capacity of the dataset has, the less likely the overfitting will occur, and the better generalization and robustness of the model will be. However, due to overwhelming time and effort cost, it is difficult to have relatively abundant samples with pre-known annotations which may limit the improvement of the detection accuracy of the CNN model to some extent. Transfer learning (Weiss et al., 2016) could be adopted to facilitate the convergency speed and improve the model's robustness especially when the customized dataset is similar or partially overlapping with the open-source dataset. Official YOLO V3 was trained based on the COCO dataset (Lin et al., 2014) which contents more than 80 classes among which is the PERSON class. Therefore, the official weights were utilized in the training process of the model involved in this research to improve its predication accuracy and convergency speed.

The training process is separated into two steps. First, import the official pre-trained weights and freeze the last three convolutional layers, iterate 200 epochs with the initial learning rate of 10^{-3} and batch size of 32. Next, unfreeze the last three convolutional layers, iterate 200 epochs again with the initial learning rate of 10^{-4} and the batch size of 8 to finetune the model. During the training process, we use ReduceLROnPlateau callback function to multiply the learning rate by a constant of 0.5 as long as the training loss stops to decline in 10 consequent iterations. Meanwhile, the Tensorboard callback function is applied to dynamically observe and save the model parameters

3.3. Depth information extraction

To eliminate noises, data of depth image has been optimized by Spatial Edge-Preserving Filter and Holes Filling Filter (referred to the Intel RealSense SDK 2.0) before depth information extraction. Considering that the gray value of each pixel of the depth image is linearly related to the distance, then the concrete distance between the target object and the camera can be extracted by picking the gray value of the pixel at the specific position. Based on the method of object detection elaborated in section 3.2, the most intuitive and reliable position is the center of bounding box (expressed as P_c below). In some cases, taking the depth data at P_c as the desired distance could be practical. However, taking into account the uncertainties of object's attribute as well as environmental condition, there are some undesirable cases in which the bounding boxes are larger, smaller, offset or even failed (as shown in Fig.2). In addition, one of the limits of YOLO V3 is that it can only output rectangular bounding box. This means that it is sensitive to image distortion. While, as the RGB-D camera is mounted on the UAV, object can be slanted in the color image because of the dynamic change of the UAV's attitude. In this case, depth data at P_c could become unreliable or even invalid. To remedy this defect, we pay additional attention to the object's gravity center, expressed as P_g, by performing local image processing on the area surrounded by the bounding box.



(a) (b) (c)

FIGURE 2. Three scenarios demonstrating the relative position between P_c and P_g. (a) Normal case. (b) The predicted bounding box of the object slants in image. (c) The bounding box is larger than the one in ideal case. From top to bottom, they are color images, depth images, gray images inside the bounding boxes and binary images with contours of object. The blue point and red point in binary image indicate P_c and P_g respectively.

This study customizes a specific strategy to improve the accuracy and reliability of the depth information acquisition considering the differences of P_c and P_g. This strategy is detailed as follows: when the variation of two points in both height and width directions under image pixel coordinates is less than 5 pixels, the average depth data at P_c and P_g will be considered as the true value; when the variation surpasses 5 pixels, the depth data at P_g will be seen as the real value; when the extraction of P_g fails, the depth data at P_c is regarded as the real value; when the depth data at P_c is void, then, the average value of all valid data on horizontal centerline of the bounding box will be adopted.

3.4. RGB-D information fusion

In order to simplify the calculation process, we establish three assumptions: (i). the intrinsic parameters of the camera are pre-known; (ii). the imaging plane of the camera is parallel to the scene plane of the object; (iii). the optical axis is inward through the center of the image plane. Under the above assumptions, the real-scene spatial coordinate information of any point selected from the image plane can be calculated following (1). This formula is derived from the principle of pinhole imaging.

$$\begin{cases} x_s = \dfrac{z \times p_x \times p_s}{f} \\ y_s = \dfrac{z \times p_y \times p_s}{f} \\ z_s = z \end{cases} \tag{1}$$

where, z is the vertical distance between the scene and the camera; f is the focal length of the color camera; p_x is the number of pixels in the horizontal direction of the image plane relative to the optical axis; p_y is the number of pixels in the vertical direction; p_s is the physical size of pixels of the color camera; x_s, y_s, and z_s are the spatial coordinates of the specific point in real scene plane.

After obtaining the coordinates of each vertex of the bounding box, the width and height of the object could be obtained following (2).

$$\begin{cases} w_o = |x_{ur} - x_{ul}| \\ h_o = |y_{ll} - y_{ul}| \end{cases} \tag{2}$$

where, w_0 is the width of the object; x_{ur} is the X-axis of upper-right vertex of the bounding box; x_{ul} is the X-axis of upper-left vertex; h_0 is the height of the object; y_{ll} is the Y-axis of lower-left vertex; y_{ul} is the Y-axis of upper-left vertex.

3.5. Obstacle avoidance strategies

Exclusive and specific collision avoidance strategies should be adopted according to the results of object detection and RGB-D information fusion since different kinds of obstacles pose distinct extents of risks to drone's flight safety. Environmental sensing method based on deep learning and the Intel RealSense D435 depth camera can simultaneously perform object detection and 3D information acquisition. However, because of light condition change, obstacle's attribute difference, and the depth camera's measurement range limit, there are some situations in which the target category and depth information cannot be acquired at the same time. The detailed analysis is as follows:

If there are no obstacles on the flight path or the obstacles are far away, no information will be obtained through object detection and RGB-D information fusion. When some obstacles appear ahead, but the distances exceed the depth camera's sensing range, then, only their categories would be available. When the distances are within the sensing range and the main contours of these obstacles can be presented within the field of view (FOV), then, their categories, spatial positions, and profiles can be obtained through our solution simultaneously. When obstacles are too close that their images completely fill the FOV, the distance information from depth image could be unreliable, and it is usually difficult to identify their categories.

The FOV is delimited into four parts which could be listed from far to near as clear area, warning area, action area and emergency area, as shown in Fig.3. In detail, clear area means there are no obstacles in front, and it is safe to keep flying with current flight parameters; in warning area, the drone can sense potential collision risks ahead, but has no knowledge of where it is, it just remembers the category of the potential obstacle and reduces flight speed if necessary; the action area is defined as the region where the drone would take specific obstacle avoidance actions according to concrete attributes of obstacles; if the obstacle appears in emergency area, the drone would stop and hover at current position immediately and wait for the intervention by pilot.

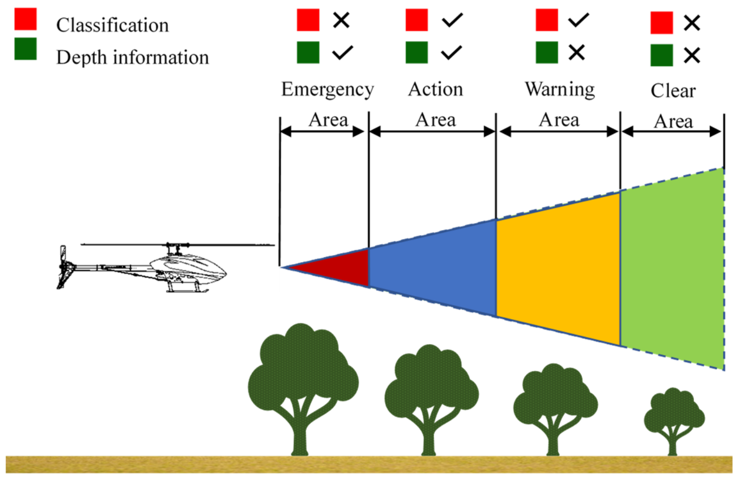


FIGURE 3 **FOV division result considering the sensing range limit of depth camera and the outcomes of object detection.**

In this section, we define some optimal collision avoidance strategies in advance according to the results of object detection when obstacles are in action area. In detail, for short small trees or buildings, the drone will not adjust flight direction, only change flight altitude to cross the obstacle; for tall and large trees or powerline poles/towers, it will turn left or right to avoid obstacles while maintaining current flight altitude; when a person or drone appears on the drone's flight path, it will immediately hover at current position and send alarming messages to the pilot.

Scattered trees in the field are the most common obstacles causing collisions risks for the drone. Therefore, taking a single tree as example, we explicitly illustrate a method about how to calculate the relative position between the tree and drone and then predict the optimal collision avoidance strategy in the light of the results of objection detection and RGB-D information fusion. As shown in Fig.4, an image plane is represented by a rectangular which has been split into four quadrants homogeneously. We set the origin of image coordinate system (X_i, Y_i) at the upper left vertex of the image, while the origin of UAV's body coordinate system (X_d, Y_d, Z_d) at the center. The positive direction of X_d is consistent with X_i, and Z_d points to the negative direction of Y_i. Y_d, identical to the forward flight direction of UAV, is indicated by the vertical inward at the image center. The red dotted rectangular with side length of 2m is deemed to be the minimum safely-passing-area.

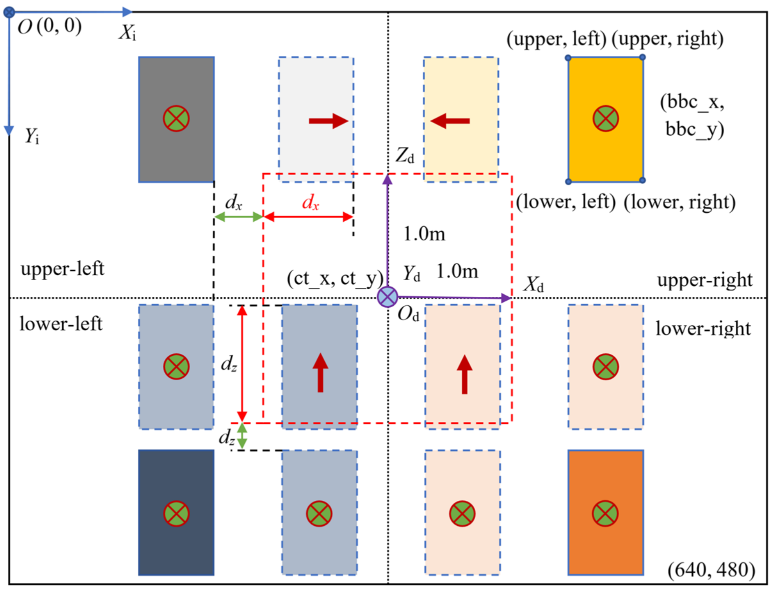


FIGURE 4 **Principles for selecting obstacle avoidance strategies according to the location of the object's bounding box in the image coordinate system. The red arrows denote the flight direction according to the corresponding collision avoidance strategies, while the cross sign indicates a risk-free obstacle. d_x is the relative distance between the left (or right) boundary of minimum safely-passing-area and right (or left) boundary of bounding box; d_z is the relative distance between the lower boundary of minimum safely-passing-area and upper boundary of bounding box. Both d_x and d_z present the relative position in UAV's body coordinate system.**

When the center of the bounding box of a tree locates in the upper-left area, then the distance between the bounding box's right border and the safely-passing-area's left border, marked as d_x, can be extracted according to (1). If d_x is positive, the UAV would ignore the existence of tree and continue to execute flight mission with the current flight parameters. If d_x is negative, the UAV would turn right with a certain distance to detour the tree. Similar obstacle avoidance strategy is also applicable when the center of the bounding box is in upper-right area. When the center of the bounding box is in lower-left area, the d_z which represents the distance between the bounding box's top border and the safely-passing-area's bottom border would also be calculated. If it is positive, the drone would pass directly, otherwise, the d_x would be regarded as the main basis for determining whether there are collision risks or not. When d_x is positive, then the obstacle is beyond the safely-passing-area. If d_x and d_z are both negative, the UAV would leap forward with a certain distance to bypass the tree. Similar obstacle avoidance strategy is also applicable to the circumstance in which the bounding box center locates in lower-right area. It is worth noting that although the horizontal displacement and leap forward obstacle avoidance strategies are applicable when the boundary box of the obstacle locates in the lower area of the image plane, we still prefer the leap forward strategy. The main reason is it generates much less instability comparing with horizontal displacement strategies. This benefits from the fact that leap forward strategy only involves the change of flight altitude, but not has the change of attitude which is the main cause of the sway of pesticide solution in the tank. d_x and d_z are two important parameters for flight path planning (described in detail in 3.6), which represent horizontal and vertical displacement respectively.

We present two examples in Fig.5. The first example demonstrates the predicted bounding box center located at the upper-left area. The orientation of the red line suggests that turn-right collision avoidance strategy is adopted, and the closer the tree is to the drone, the longer the red line is. The second instance indicates the case in which the tree locates at the lower-right area and the leap forward collision avoidance strategy should be executed.

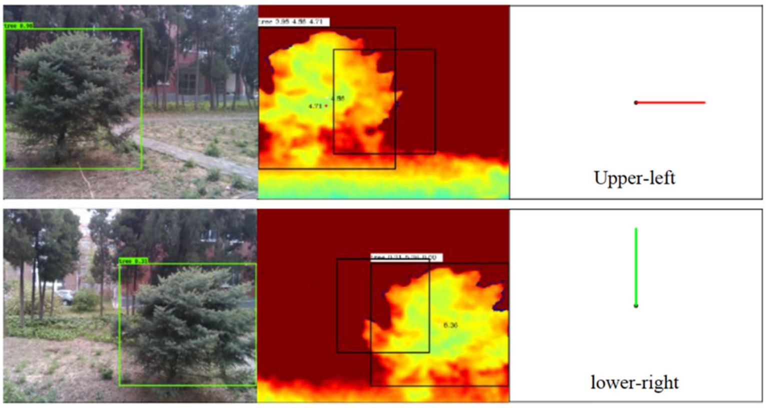


FIGURE 5 Specific obstacle avoidance strategies in two example scenarios where the centers of predicted bounding boxes located at upper-left and lower-right area in the image coordinate system respectively.

3.6. Flight path planning

After the obstacle avoidance strategy have been explicated, the next question is how to generate an optimal collision avoidance trajectory to minimize the adverse impact on the effectiveness and efficiency of the drone. Figure.6 depicts how to calculate the offset under three circumstances in which right turn, left turn and leap forward detouring strategies are needed to be implemented respectively.

To make it clear, in this section we take the left turn obstacle avoidance strategy to avoid a single tree as an example. Geodetic coordinate system and UAV airframe coordinate system are established separately in order to describe the relative position between the UAV and the tree. The origin of the geodetic coordinate system O_e is located at the starting point of the UAV's flight task, with X_e pointing to the East and Y_e pointing to the North. The origin of the UAV's airframe coordinate system is located at the center of gravity, as X_d representing the right side of the UAV and the Y_d pointing to the forward flight direction. Both Z_e and Z_d are coincident with the direction of increasing altitude. In order to simplify the generation of collision avoidance path and clearly illustrate the method of calculating the coordinates of task-points, we proposed some hypothesis or preconditions:(i). The obstacles exist independently; (ii). The outer contour of the cross section of the obstacle is round; (iii). The starting position of obstacle avoidance task is 2m away from the obstacle; (iv). Following the principle of minimizing the total distance during obstacle avoidance task.

As shown in Fig.6-a, supposing that the drone is performing a multi-task-points straight-line flight mission in the direction of O_eP_0. When it reaches point P_0, the single tree enters the action area where its classification, height, width and position can be obtained at the same time. Then point $P_1(x_1, y_1, z_1)$ that is 2m from the tree is defined as the starting point of the obstacle avoidance task. In addition, the coordinates of P_2 (x_2, y_2, z_2) and P_3 (x_3, y_3, z_3) could be computed following (3) and (4) which are derived through geometric relations. Based on the straight-line flight capability, the UAV performs obstacle avoidance trajectories composed of P_0, P_1, P_2, and P_3, and resumes the straight-line flight mission after the obstacle avoidance mission is completed. Similarly, the coordinates of P_2 and P_3 can be obtained following (5)-(6) or (7)-(8) when it is needed to execute right-turn or leap forward collision avoidance strategies. The corresponding collision avoidance paths are shown as Fig.6-b and Fig.6-c.

$$\begin{cases} \boldsymbol{\beta = tan^{-1}\left(\frac{d_x}{d}\right)} \\ \boldsymbol{\theta = \alpha - \beta} \\ \boldsymbol{d = 2 + w_o/2} \\ \boldsymbol{x_2 = x_1 + \sqrt{{d_x}^2 + d^2} \times sin\,\theta} \\ \boldsymbol{y_2 = y_1 + \sqrt{{d_x}^2 + d^2} \times cos\,\theta} \\ \boldsymbol{z_2 = z_1} \end{cases} \tag{3}$$

$$\begin{cases} \boldsymbol{\theta = \left(\frac{\pi}{2} - \alpha\right) - \beta} \\ \boldsymbol{d = 2 + w_o/2} \\ \boldsymbol{x_3 = x_2 + \sqrt{{d_x}^2 + d^2} \times \cos\theta} \\ \boldsymbol{y_3 = y_2 + \sqrt{{d_x}^2 + d^2} \times \sin\theta} \\ \boldsymbol{z_3 = z_2} \end{cases} \tag{4}$$

$$\begin{cases} \boldsymbol{\beta = \tan^{-1}\left(\frac{d_x}{d}\right)} \\ \boldsymbol{\theta = \left(\frac{\pi}{2} - \alpha\right) - \beta} \\ \boldsymbol{d = 2 + w_o/2} \\ \boldsymbol{x_2 = x_1 + \sqrt{{d_x}^2 + d^2} \times \cos\theta} \\ \boldsymbol{y_2 = y_1 + \sqrt{{d_x}^2 + d^2} \times \sin\theta} \\ \boldsymbol{z_2 = z_1} \end{cases} \tag{5}$$

$$\begin{cases} \boldsymbol{\theta = \alpha - \beta} \\ \boldsymbol{d = 2 + w_o/2} \\ \boldsymbol{x_3 = x_2 + \sqrt{{d_x}^2 + d^2} \times \sin\theta} \\ \boldsymbol{y_3 = y_2 + \sqrt{{d_x}^2 + d^2} \times \cos\theta} \\ \boldsymbol{z_3 = z_2} \end{cases} \tag{6}$$

$$\begin{cases} \boldsymbol{d = 2 + w_o/2} \\ \boldsymbol{x_2 = x_1} \\ \boldsymbol{y_2 = y_1 + d} \\ \boldsymbol{z_2 = z_1 + d_z} \end{cases} \tag{7}$$

$$\begin{cases} \boldsymbol{d = 2 + w_o/2} \\ \boldsymbol{x_3 = x_2} \\ \boldsymbol{y_3 = y_2 + d} \\ \boldsymbol{z_3 = z_2 - d_z} \end{cases} \tag{8}$$

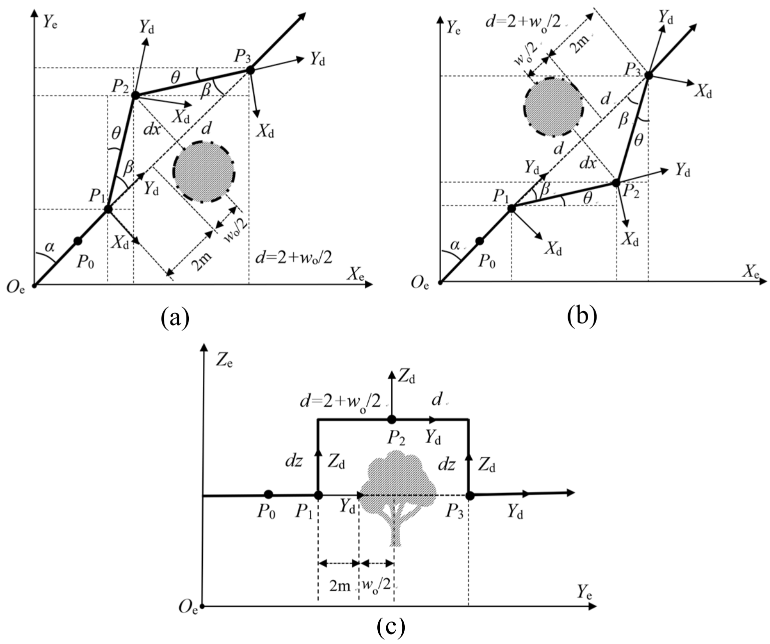


FIGURE 6 Geometric analysis for generating the avoidance paths when taking left turn (a), right turn (b) and leap forward (c) avoidance strategies.

3.7. Autonomous flight control method

For the sake of maintaining the expansion flexibility of the entire flight control system without compromising its reliability and stability, the two-tier control system including a companion computer and a flight control system is proposed. Companion computer running ROS acts as the main-controller and flight control system acts as the sub-controller. Specifically, the main-controller with abundant peripheral interfaces is responsible for executing high-level control procedures such as real-time data acquisition, image processing, inference of CNN, and generation of attitude and position control commands for the UAV. It communicates with other devices that support ROS through the mechanisms of Topic and Service. Due to the sustainable contribution from the open-source community, ArduPilot has been proved to be a reliable flight control firmware for the innovation and implementation of personalized application based on the UAV platform. The sub-controller companioned with ArduPilot, as an independent flight controller, adjusts the drone's attitude according to the commands received from main-controller, ground-station or remote controller via messages in MAVLINK protocol and broadcasts its real-time state parameters in the opposite direction. MAVROS acts as a bridge connecting companion computer and flight controller by shouldering the responsibility to do bidirectional conversion between ROS and MAVLINK messages. This autonomous flight control method integrates the flight control system, companion computer and Intel RealSense D435 into a seamless system.

In this work, we focus on the spatial position control of the UAV by sending corresponding commands and 3D position coordinates to the flight controller who completes attitude control through bottom driver. In order to simplify the flight task, we divide it into four independent subtasks: takeoff, flight straightly towards the task point, hover for a specific time and autonomous landing. A common flight mission can be generated by freely combining these four subtasks. Based on ROS and MAVROS, the flight mission management system, we called TCS, is developed. It not only assumes the duty of maintaining the stability of communication inside the two-tires control system but also completes the scheduling of different flight missions by continuously querying the execution progress of each subtask and the entire task.

4. Experiments and Results

The contents of our experiments are composed of three sections. Firstly, an experiment was conducted to evaluate the performance of the obstacle detection CNN model with the validation dataset. Secondly, to assess the sensing range of depth camera and the predication accuracy of objects profile and 3D spatial position, a real-world test was carried out. Thirdly, we launched an experiment that combines simulation environment with real object to examine the UAV’s comprehensive capacities, including environmental perception, obstacle avoidance and autonomous flight.

4.1. Performance of CNN model

Object detection accuracy, interference speed and generalization ability are three important indicators that reflect the performance of the CNN model. For our proposed solution, both the classification accuracy and bounding box predication accuracy influence the precision of RGB-D information fusion directly. In this study, we use Detection Accuracy (DA) which represents the product of the two to assess the performance of the CNN model.

We launched a series of repeated experiments with the validation dataset to evaluate the detection accuracy as well as to assess the interference speed. The details of test results are present in Tab. 1. Results suggest that the average precision (AP) of each category exceeds 90% except Power-line Pole/Tower. This is because we classified power-line poles and power-line towers into the same category, although there are significant differences in their shape features. Nevertheless, the mAP (means of APs) of the five classes reaches 91.9% which shows that the CNN mode has strong generalization ability. DA of each category is 74.3%, 77.8%, 66.0%, 72.2% and 86.9% respectively. The average DA of the five categories is 75.4%. Additionally, the average time cost for detecting single image is about 53.33ms which means it can update the results of environmental perception to the drone more than 18 times per second without considering the communication delay.

Figure 7 shows some object detection results in the validation dataset. It can be found that the predicted bounding boxes can surround the obstacles precisely with high confidences.

Table 1 Results of object detection tests

	Person/%	Tree/%	Power-line Pole/Tower/%	Building/%	Drone/%
AP	92.4	92.2	87.9	90.3	96.7
IoU	80.4	84.4	75.1	79.9	89.9
DA	74.3	77.8	66.0	72.2	86.9

Note: AP stands for average classification precision for each class; IoU represents the predicting accuracy of bounding box; DA indicates the Detection Accuracy combining the AP and IoU.

FIGURE 7. Examples of object detection results.

4.2. Accuracy of RGB-D information fusion

Taking single tree (growing on the campus of China Agricultural University, Beijing, 100083, China) whose real width is 3.20m and real height is 2.85m as an example, we conducted a real-world experiment to investigate the prediction accuracy of profile and position based on RGB-D information fusion. In this experiment, 14 sampling points with a step length of 0.5m from the starting point (2.5m away from the center of the trunk) to the end point (9.00m away from it) are set up. These parameters are determined according the reliable sensing range of Intel RealSense D435 Depth Camera. Each sampling-point’s color and depth images are presented in Fig.8, and the corresponding results of object detection and RGB-D information extraction are shown in Fig.9.

As shown in Fig.8, when the relative distance between the tree and the camera is less than 4.5m, measurement errors of the tree’s width and height are relatively large. This is because the tree is only partially visible. As the relative distance increase, the complete image of the tree can be included in the color image. When it is between 4.5m and 8.0m, the image of the tree can be seen in both color images and depth images, and the results of object detection and depth data extraction would be trustful. When the relative distance is greater than 8.0m, the deep learning algorithm still can effectively predict the tree’s category and bounding box although the target tree occupies a small area in the color image. However, depth data accuracy deteriorates gradually as it becomes hard to effectively distinguish the tree and background in the depth image.

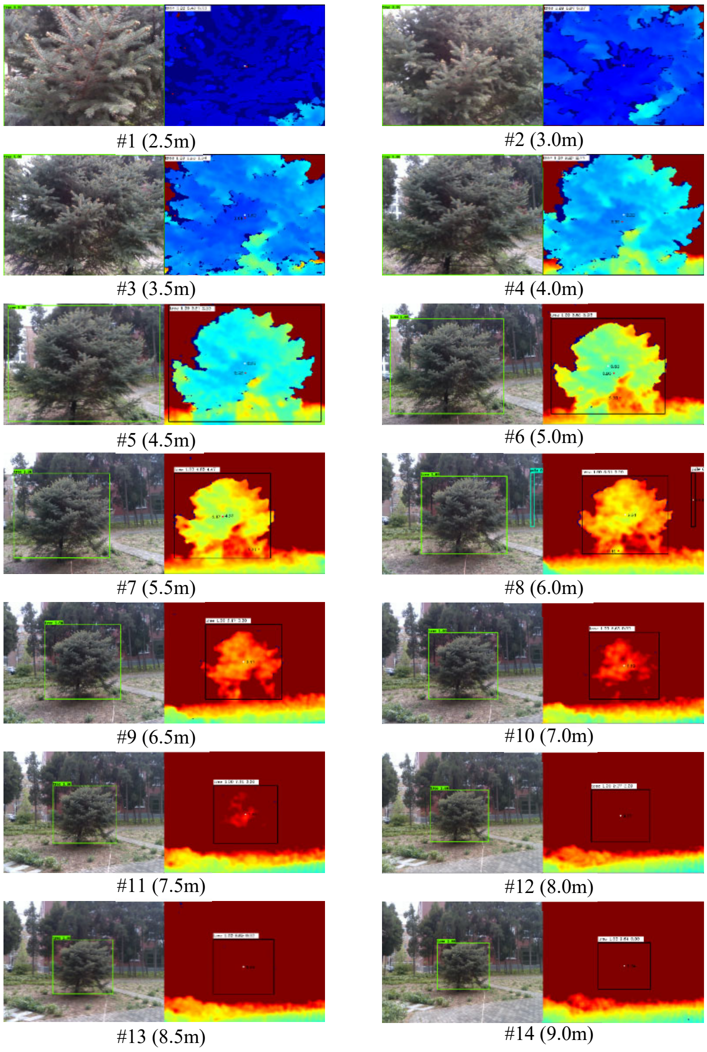


FIGURE 8. Color and depth images of each sampling-point from #1 to #14.

More details about the performance of object detection and RGB-D information extraction can be found in Fig.9. We use Confidence Score (CS) to represent the probability that the model predicts the target category as a tree. The average CS of the 14 sample-points is 0.99. This means that the change of distance has little effect on the accuracy of deep learning object detection. In terms of the results of RGB-D information extraction, the average error of depth data, width and height is -0.77m, -0.67m and -0.65m respectively. However, when the camera is between 4.5m (sampling-point 5) and 8.0m (sampling-point 12) away from the trees, the errors are -0.53m, -0.26m and -0.24m separately. The results indicate that the measured data is generally smaller to the true value.

Particularly, from sampling-point 1 to 10, the error of depth data is negative, but its absolute value decreases with the increase of distance. From sampling-point 11 to 14, the error of depth data becomes positive, and it increases in line with the increase of distance. Additionally, from sampling-point 1 to 8, the prediction errors of width and height are negative, and its absolute value decreases as the distance increase. While, from sampling-point 9 to 14, the prediction errors of both width and height fluctuate little, and their average errors stabilize at -0.05m and 0.03m respectively.

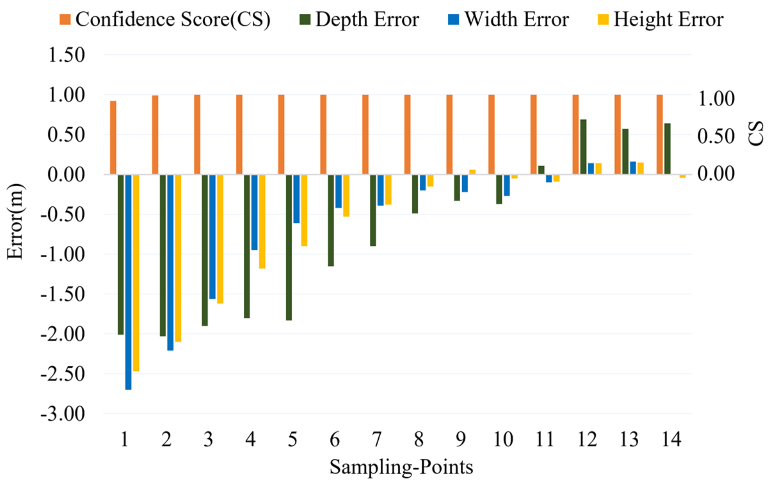


FIGURE 9. Results of RGB-D information extraction.

To sum up, the prediction accuracy of profile and location of the object depends heavily on the relative distance between the camera and object. Specifically, when the relative distance is 7.5m, object detection precision and 3D information acquisition performance can reach the optimal state at the same time.

4.3. Simulation flight experiments

In order to verify the UAV's abilities of environmental perception, collision avoidance, and autonomous flight, we proposed a safe and effective method as combining the simulation environment with the real world. In the simulation environment, a virtual UAV was controlled by the TCS to execute a straight-line flight mission. Meanwhile, customized scripts assume the burden of sensing the surrounding environment and generating the avoidance strategy and flight path when necessary. In real world, we used the Intel RealSense D435 to feed the color and depth images of a real single tree into the CNN model. When the prediction results suggest that the tree is on the flight path and there are potential collision risks, the UAV will automatically interrupt the current straight-line flight mission and perform the obstacle avoidance procedure. After bypassing the tree in the simulation environment, the UAV will automatically resume former straight-line flight mission. During the tests, we manually adjust the FOV of Intel RealSense D435 to trigger the left-turn, right-turn or leap forward collision avoidance procedure respectively.

The results of object detection, 3D spatial position extraction, profile prediction and the whole flight trajectories under three different circumstances are comprehensively presented in Fig.10. It can be found that the simulated flight trajectories are consistent with the anticipate tracks which have been introduced in Fig.5. The experimental results showed that the proposed solution can automatically control the UAV to perform autonomous flight and obstacle avoidance tasks according to the obstacle's specific attributes.

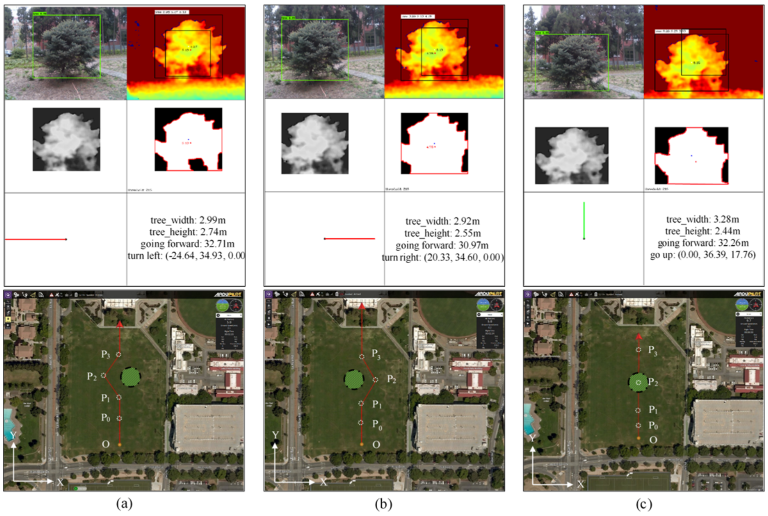


FIGURE 10. Simulation results of avoidance strategy and flight path under three different circumstances. (a): Left-turn. (b): Right-turn. (c) Leap forward. The tree size and calculated position offset have been deliberately magnified by 10 times in order to make the flight trajectory clearer.

5. Discussion

Based on the experimental results, to some extent, our proposed systems and methods of environmental perception, collision avoidance and autonomous flight control have improved the UAV's automation level and flight safety.

Having the knowledge of what the ahead obstacle is fundamental but important for the UAV's flight safety and working efficiency. Comparing with traditional methods of obstacle detection, we introduced a depth camera to sense the flight environment with higher information dimensions. The state-of-the-art deep-learning based object detection algorithms was adopt to understand the color images of the real flight scene. Object detection results indicated the CNN model can precisely predict the obstacle's category and bounding box with the AP of 91.9% within 53.33ms. Although the precision and speed maybe not good enough in some rigorous conditions, but it has significantly improved the plant protection UAV's environmental perception abilities given the facts that the categories of obstacles in farmland are generally definite and their distributions are relatively independent.

Object's profile and 3D spatial position can be extracted by fusing the RGB-D information. However, test results suggested that the measuring errors is not a constant, but a dynamic value. This phenomenon could be caused by many reasons, such as the distance between the camera and the object, the limits of the sensing rang of the depth camera, the changes of light intensity, the differences of observation angle, etc. In this work, the errors of RGB-D information extraction can reach the minimum when the distance is 7.5m. Nevertheless, this distance is very valuable for the drone to take appropriate measures to avoid collision when obstacles appear, especially considering the fact that the normal flight speed of the plant protection drone is generally less than 5m/s.

Although not considering the influence of many practical factors, the simulation results still verified the effectiveness of our proposed solution. By applying a depth camera and deep learning, the drone can avoid obstacles autonomously based on the knowledge of obstacle's attributes.

6. Conclusion

In this paper, a novel solution for enhancing the UAV's environmental perception and autonomous obstacle avoidance abilities was proposed. Taking advantages of deep-learning based object detection algorithm and Intel RealSense D435 depth camera, we introduced a new tactic to obtain the obstacle's classification, profile and 3D spatial position via comprehensively integrating RGB-D information. According to the obstacle's specific properties, we elaborated the methods of generating the optimal collision avoidance strategy and planning the distance-minimized flight path. Besides, customized scripts and TCS were developed to improve the UAV's autonomous flight capability. For evaluating the performance of presented solution, a series of experiments were carried out. Results indicated that DA of CNN model is 75.4% and it costs about 53.33ms for processing single. Additionally, when the camera is between 4.5m and 8.0m away from the tree, the errors of depth data, width and height are -0.53m, -0.26m and -0.24m respectively. Comprehensive simulation flight experiment implied that our proposed solution can significantly improve the UAV's environmental perception, obstacle avoidance and autonomous flight abilities. Furthermore, this study is helpful to promote the implements of UAVs in broader applications.

However, there are still some limitations of this work particularly when considering the complexity of unstructured farmland environment, the dynamically changing environmental parameters and the robustness of the control algorithms. In the future work, we will continuously optimize of details of our solution and make it more applicable in actual applications.

Acknowledgments

This work was supported by The National Key Research and Development program of China (2016YFD0200700).

References

Adrian, G., Daniel, C., Manuel, M., Alberto, B., 2020. Autonomous navigation for UAVs managing motion and sensing uncertainty. Robot. Auton. Syst, 126. DOI: 10.1016/j.robot.2020.103455.

Ahmad, F., Qiu, B., Dong, X., Ma, J., Huang X., Ahmed, S., Chandio, FA., 2020. Effect of operational parameters of UAV sprayer on spray deposition pattern in target and off-target zones during outer field weed control application. Comput. Electron. Agric.172. DOI: 10.1016/j.compag.2020.105350.

Basso, M., de Freitas, EP., 2020. A UAV guidance system using crop row detection and line follower algorithms. J. Intell. Robot. Syst. 97(3-4),605-621. DOI: 10.1007/s10846-019-01006-0.

Feng, A., Zhou, J., Earl D, V., Kenneth A, S., Zhang, M., 2020. Yield estimation in cotton using UAV-based multi-sensor imagery. Biosyst. Eng. 193, 101-114. DOI: 1 10.1016/j.biosystemseng.2020.02.014.

Fernando, V., Dmitry, B., Kevin, P., John, W., Flipe G., 2018. A novel methodology for improving plant pest surveillance in vineyards and crops using UAV-based hyperspectral and spatial data. Sensors. 18(01), 460-466. DOI: 10.3390/s18010260.

Floreano, D., Wood, R.J., 2015. Science, technology and the future of small autonomous drones. Nature. 521, 460-466. DOI: 10.1038/nature14542.

Jongho, P., Namhoon, C., 2020. Collision Avoidance of Hexacopter UAV Based on LiDAR Data in Dynamic Environment. Remote Sens. 12(06). DOI: 10.3390/rs12060975.

Liao, J., Zang, Y., Luo, X., Zhou, Z., Lan, Y., Zang, Y., Gu, X., Xu, W., Hewitt, A., 2019. Optimization of variables for maximizing efficacy and efficiency in aerial spray application to cotton using unmanned aerial systems. *Int. J. Agric. Biol. Eng.* 12(2), 10-17. DOI: 10.25165/j.ijabe.20191202.4288.

Lin, T., Marie, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dolla, P., Zitnick, C., 2014. Microsoft COCO: Common Objects in Context. *ECCV 2014. Lecture Notes in Computer Science*. 8693, 740-755. DOI: 10.1007/978-3-319-10602-1_48.

Liu, Z., Shi, S., Duan, Q., Zhang, W., Zhao, P., 2019. Salient object detection for RGB-D image by single stream recurrent convolution neural network. *Neurocomputing*. 363, 46-57. DOI: 10.1016/j.neucom.2019.07.012.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.C., 2016. SSD: Single Shot MultiBox Detector. in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. 21-37.

Loghmani, M.R., Plananmente, M., Caputo, B., Vincze, M., 2019. Recurrent convolutional fusion for RGB-D object recognition. *IEEE Rob. Autom. Lett.* 4(3), 2878-2885. DOI: 10.1109/LRA.2019.2921506.

Minaeian, S., Liu, J., Song, Y., 2018. Effective and efficient detection of moving targets from a UAV's camera. *IEEE Transactions on Intelligent Transportation Systems*. 19(2), 497 - 506. DOI: 10.1109/TITS.2017.2782790.

Mohta, K., Watterson, M., Mulgaonkar, Y., Liu, S., Qu, C., Makineni, A., Saulnier, K., Sun, K., Zhu, A., Delmerico. J., Karydis, K., Atanasov, N., Loianno, G., Scaramuzza, D., Daniilidis, K., Taylor, C., Kumar, V., 2018. Fast, autonomous flight in GPS-denied and cluttered environments", *J. Field Robot.* 35(1), 101-120. DOI: 10.1002/rob.21774.

Perez-Grau, F., Ragel, R., Caballero, F., Viguria, A., Ollero, A., 2018. An architecture for robust UAV navigation in GPS-denied areas. *J. Field Robot.* 35(1), 121-145. DOI: 10.1002/rob.21757.

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *in Proc. IEEE. Int. Conf. Compu. Vis. Patt. Recog. (CVPR),* Workshops, Salt Lake City, UT. 625-633.

Shao, S., Peng, Y., He, C., Du, Y., 2018. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Trans.* DOI: 10.1016/j.isatra.2019.08.018.

Tetila, EC., Machado, BB., Menezes, GK., Oliveira, AD., Alvarez, M., Amorim, WP., Belete, NAD., da Silva, GG., Pistori, H., 2020. Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.,* 17(5), 903-907. DOI: 10.1109/LGRS.2019.2932385.

Weiss, K., Khoshgoftaar, T., Wang, D., 2016. A survey of transfer learning. *J Big Data*. 3(9). DOI: 10.1186/s40537-016-0043-6.

Xu, Y., Xue, X., Sun, Z., Chang, C., Gu, W., Chen, C., Jin, Y., Peng, B., 2019. Online spraying quality assessment system of plant protection unmanned aerial vehicle based on Android client. *Comput. Electron. Agric.* 166, 1-8. DOI: 10.1016/j.compag.2019.104938.

Xue, X., Lan, Y., Sun, Z., Chang, C., Hoffmann, W.C., 2016.Develop an unmanned aerial vehicle based automatic aerial spraying system. *Comput. Electron. Agric*. 128, 58-66. DOI: 10.1016/j.compag.2016.07.022.

Yang, Z., Xiao, Y., Yang, Z., Wu, Y., Qi, L., 2019. A real-time route planning method based on morphological segmentation for plant protection UAVs. *2019 ASABE Annual International Meeting*. DOI: 10.13031/aim.201900490.

Yann, L., Yoshua, B., Geoffrey, H., 2018. Deep Learning. Nature. 521, 436-444. DOI: 10.1038/nature14539.

Zia, S., Yuksel, B., Yuret, D., Yemez, Y., 2017. RGB-D object recognition using deep convolutional neural networks. in *Proc. IEEE. Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy. 896-903. DOI: 10.1016/j.patcog.2017.07.026.