IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 24, 2021, accepted July 14, 2021, date of publication July 19, 2021, date of current version July 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097945

Autonomous Quadrotor Navigation With Vision Based Obstacle Avoidance and Path Planning

HUEI-YUNG LIN[1], (Senior Member, IEEE), AND XIN-ZHONG PENG[2]

[1]Department of Electrical Engineering, Advanced Institute of Manufacturing with High-Tech Innovation, National Chung Cheng University, Minxiong 62102, Taiwan

[2]Department of Electrical Engineering, National Chung Cheng University, Minxiong 62102, Taiwan

Corresponding author: Huei-Yung Lin (lin@ee.ccu.edu.tw)

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 106-2221-E-194-004, and in part by the Advanced Institute of Manufacturing with High-Tech Innovations (AIM-HI) from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

ABSTRACT Due to the high demands on military and commercial applications, the development of UAVs (unmanned aerial vehicles) has become increasingly important in recent years. In this paper, we present a vision guided autonomous navigation approach for quadrotor UAVs. A map-based offline path planning technique is developed to generate an initial path, followed by the waypoints of the trajectory for flight guidance. During the navigation, an onboard camera is utilized to acquire a sequence of monocular images for environment perception. A vision-based obstacle detection technique using optical flow is proposed for collision avoidance. The optical flow field constructed from the image sequence is used to provide the depth cues for the incoming obstacle detection. A single-board computer is adopted as a control platform, and the proposed algorithms are implemented for online and real-time processing. Several experiments are carried out in the outdoor environment for obstacles avoidance and visual guidance. The results have demonstrated the feasibility of our proposed method for path planning and autonomous navigation.

INDEX TERMS Unmanned aerial robot, path planning, obstacle avoidance, machine vision.

I. INTRODUCTION

In recent years, the development and applications of UAVs (unmanned aerial vehicles) has become more and more popular. While UAVs were mainly used for the military and civil operations such as aerial inspection and terrain modeling in the past few decades, it is now growing rapidly for commercial uses [1], [2]. Due to the advances of communication, control and computing technologies, as well as the hardware cost-down, the industry has put significant efforts on making the UAV an indispensable tool in our daily life. Many companies start to provide some commercial services such as air delivery, aerial imaging [3], etc. In this IoT era, people are also able to connect to UAVs via smartphones for various routine jobs or innovative applications. By the integration with IMU (inertial measurement unit), GPS and other sensors, the realization of autonomous flight is becoming possible, but still requires further investigation [4].

According to the design principle, there are two types of UAVs commonly available in the market: fixed-wing drones and quadcopters. The fixed-wing drones are able to provide long endurance flight, but the motion control is not as flexible as quadcopters. They require a much larger radius of gyration for direction change, and are generally used for aerial imaging applications. On the other hand, the control of quadrotors is more stable, flexible, and able to provide the vertical take-off, landing and hover [5]. However, the payload and flight endurance are very limited. Nevertheless, operating remotely from the ground station would not fulfill the requirements for most recent applications. The path planning, navigation and safety issues of autonomous flight are widely discussed in the robotics community [6].

In the robotics research areas, autonomous navigation generally includes the capabilities of path planning and obstacle avoidance. One important objective is to provide an optimal path for safe navigation. For the UAV applications, an initial trajectory is usually provided for navigation guidance. The waypoints from path planning are then used to instruct the heading control of the UAV [7]. In general, the flight

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

trajectories are planned off-line, with the assistance from the geographic information system (GIS). Similar to the map-based vehicle or mobile robot navigation, UAV flight planning will generate a shortest or optimal path in the virtual environment. Different from the ground vehicles, it is possible to avoid the obstacles by creating a high altitude flight path. Nevertheless, people are more concerned with the quadrotor UAVs flying in the low altitude environment. Thus, the related safety issues need to be carefully addressed.

This paper investigates autonomous navigation of quadrotor UAVs in the real-world environment. While the quadrotors have been adopted for many applications over the years, the ability to keep the flight safe in autonomous navigation is still a great challenge [8], [9]. Due to the limited payload capabilities and low cruising endurance, it is required to use lightweight onboard devices for the UAVs [10]. Consequently, the sensors such as LIDAR or sonar are typically too large, heavy or expensive to install on quadrotors [11]. In this work, we utilize a low-cost camera system with vision-based algorithms for the obstacle avoidance. With map-based path planning prior to the autonomous navigation, the main objective is to avoid collision during the flight. This is specifically important in a low altitude flight with the obstacles not shown on the maps, such as trees, traffic lights, and moving objects [12]. More detailed information and literature review about collision detection for low-altitude flight can be found in [13].

In the existing literature, vision-based obstacle avoidance approaches for the UAV can be divided into three categories [14]. The first approach is based on the monocular visual cues. It relies on the characteristics of acquired image sequences to provide the contour or texture information of the objects [15]. The regions in the image can then be classified into obstacles or non-obstacles. In recent work, Yang *et al.* [16] proposed a lightweight probabilistic CNN (convolutional neural network) for monocular depth prediction and obstacle avoidance. The technique was integrated with a visual odometry method to improve the prediction accuracy. Their algorithm was able to run in real-time on a GPU platform. He *et al.* [17] presented a bio-inspired monocular vision perception method for obstacle avoidance of micro UAVs. Since the accurate distances were not calculate explicitly, the technique was more computational efficient. They provided mapless navigation using a local planner is trained by deep reinforcement learning.

For the stereo vision based obstacle avoidance approach, multiple cameras are used for stereo matching and depth computation [18]. In the previous work, Park and Kim proposed a collision avoidance algorithm using a stereo sensor [19]. Except for the derivation of depth information, the concept of the collision cone by considering the flight velocity was adopted for collision avoidance from the quadrotor's point of view. Oleynikova *et al.* [20] presented a lightweight on-board stereo vision system using FPGA. It was designed for MAV (micro aerial vehicles) applications, and able to run up to 60 frames per second. Alternatively, a stereo vision based avoidance strategy which allows constant speed maneuvers was proposed [21]. The approach was able to deal with non-holonomic motion constraints, and implemented on a flapping wing MAV. In earlier work, a binocular stereo method was adopted for obstacle detection [22]. It was used to develop a visual threat awareness system for the UAV's real time collision avoidance.

The third approach for obstacle avoidance is to use the motion parallax associated with the camera motion. A common implementation is the optical flow method which utilizes the motion flow obtained from a sequence of images. It can be combined with either single or multiple cameras to detect the objects or environment structures in the scene. In [23], Cho *et al.* presented an optical flow technique specifically for textured 3-D environment. The divergence at the focus of expansion was considered to deal with the wall-like frontal obstacles. McGuire *et al.* first used stereo vision to estimate the depth information of the environment [24]. The results were then combined with optical flow for the velocity computation and obstacle avoidance. There were also some works which considered the obstacle detection simultaneously for multiple quadcopters [25], [26]. Nevertheless, these techniques did not take the flight planning into account to improve the real-time computation for collision detection.

In this paper, we integrate both the online obstacle avoidance and offline path planning scheme for autonomous flight control. A map-based offline guidance technique is adopted to generate an initial path using the rapidly-exploring random tree (RRT) algorithm. It is then followed by the waypoints of the generated path for flight guidance. During the navigation, an onboard camera is utilized to acquire a sequence of images for the optical flow field construction and obstacle detection as illustrated in Figure 1. It provides the relationship between the quadrotor motion and the obstacles for navigation. Several machine vision techniques are adopted to increase the robustness of our approach. The proposed algorithms are finally implemented on a single-board computer for online and real-time processing. We have performed experiments with various environment conditions. The results have demonstrated that the proposed method is able to avoid objects and reroute to the destination.

The rest of the paper is organized as follows. First, we describe the obstacle avoidance algorithm in Section II. The description of the autonomous flight system is then provided in Section III. To evaluate the proposed method, a number of experiments are conducted, and the results are presented in Section IV. The conclusion is finally drawn in Section V.

II. OBSTACLE AVOIDANCE SYSTEM

A system flow of the proposed obstacle avoidance technique is illustrated in Figure 2. We develop the vision-based algorithms for real-time image processing and analysis. It consists of several stages, starting from the image acquisition, pre-processing, optical flow computation, obstacle detection, and the final decision making. In general, the optical flow based

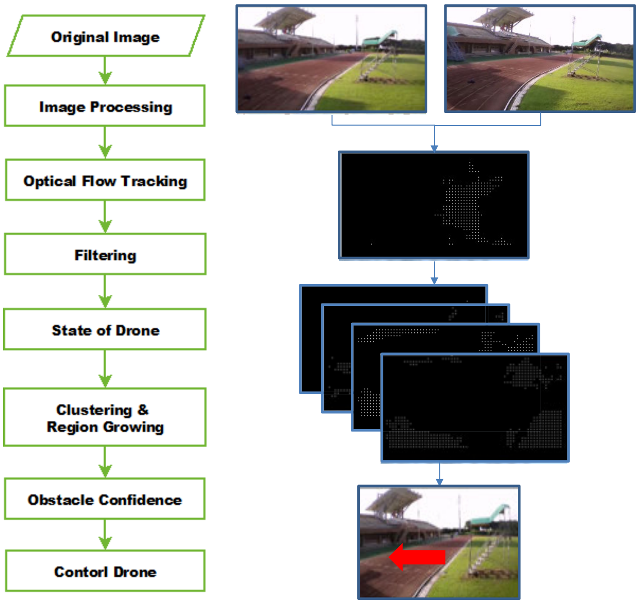


FIGURE 1. An illustration of the proposed vision-based obstacle avoidance technique using optical flow. It consists of several stages, starting from the image pre-processing, optical flow computation, region segmentation and object detection, to the final decision making.

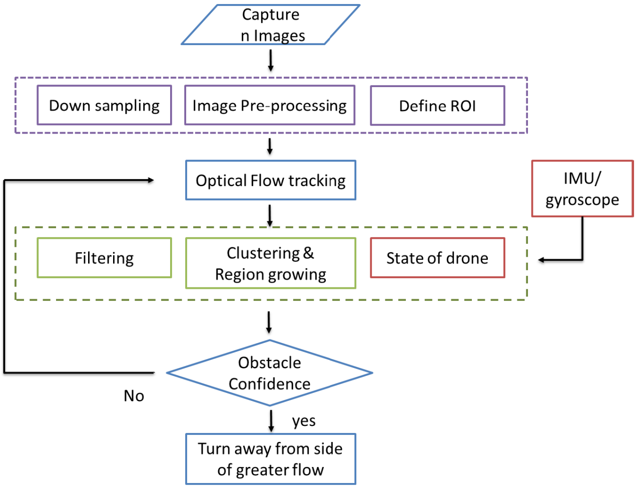


FIGURE 2. The system flowchart of the proposed obstacle avoidance algorithm. In additional to the visual information, the IMU data are also used to check the quadrotor state for flight control.

approaches are difficult to achieve the real-time performance on the low-cost hardware. Thus, one of the main objectives of the proposed technique is to reduce the computational cost in various situations.

In the pre-processing stage, the acquired image sequence is first down-sampled to reduce the computational load for the subsequent processing steps. The images are then converted to grayscale, and Gaussian filtering is carried out for noise removal. Furthermore, only the central region of the image needs to be considered for obstacle avoidance if the camera's field-of-view (FOV) is sufficiently large. It should be noted that when the quadrotor UAV proceeds forward, there will be a tilt angle facing downward for the pose. Thus, the image captured during the quadrotor motion as shown in Figure 3(b) is different from the image obtained statically (as shown in Figure 3(a)). Consequently, the region-of-interest (RoI) in the implementation is given by the red bounding box as shown in Figure 5(b) for obstacle avoidance. The overall computation can then be further reduced.

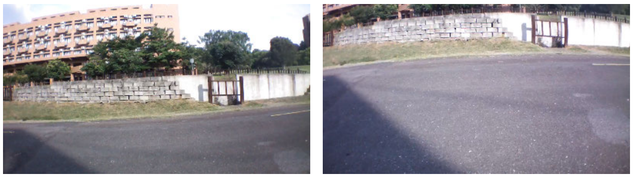

(a) Static image capture. (b) Image capture during motion.

FIGURE 3. (a) The image captured statically. (b) The image captured during the quadrotor forward motion.

To detect the changes in an image, the optical flow methods are commonly used techniques. The optical flow in the image is used to identify the motion vectors induced by the relative motion between the camera and the environment. In this work, the dense optical flow technique is adopted to calculate the dynamic change in the scene [27]. Compared to sparse optical flow methods [28], it is able to calculate the displacement of each pixel between two consecutive image frames extensively. Our optical flow estimation is developed based on Farnebäck's approach [29]. It consists of the assumptions of brightness constancy and smoothness constraint [30]. This method approximates the displacement of neighboring pixels using a quadratic polynomial expansion, and provides better results in terms of computation speed and accuracy. However, there are still restrictions for its use on the obstacle avoidance algorithm. Some issues include the optical flow vanished due to the slow camera movement, sensitive to the leaf trembling, illumination change and background noise in outdoor scenes. Thus, it is necessary to combine with other techniques for the motion analysis of potential obstacles.

In the optical flow method [31], the images are acquired from the camera continuously, and can be treated as a function of time: $I(t)$. The brightness of a pixel (x, y) at time t can be expressed by $I(x, y, t)$. That is, the image is treated as a function of time and position. Its brightness range is the intensities of the pixels in the image. If we consider a 3D scene point with the corresponding image point at (x, y), the image coordinates will change when the camera is in motion. The projection of the 3D scene point in the image is calculated at other time instants, while the pixel brightness value of the same spatial point is fixed among the image sequence. Given a pixel at the position (x, y) at time t, we assume that it moves to the position $(x + dx, y + dy)dt$ at time $t + dt$. Since the brightness remains the same, we will have

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \tag{1}$$

In the above equation, the constant brightness assumption is strong and not always satisfied in the real situations. The image intensity might be different due to the surface material of the object. Moreover, the camera exposure adjusted with automatic settings will make the image darker or brighter. Nevertheless, it still provides sufficient visual cues for the outdoor environment with ambient light.

Applying Taylor expansion, the optical flow equation (1) can be approximated by

$$I(x+dx, y+dy, t+dt) \approx I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \quad (2)$$

Under the constant brightness assumption, the intensity value remains unchanged over time, so we have

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial I}{\partial t} \quad (3)$$

where dx/dt and dy/dt are the displacements of a pixel in the x and y directions per unit time and denoted by u and v, respectively. Similarly, $\partial I/\partial x$ and $\partial I/\partial y$ are the image gradients in the x and y directions and denoted by I_x and I_y, respectively. Let I_t be the brightness change of the image with respect to time. Then, Eq. (3) can be written in the matrix form

$$\begin{pmatrix} I_x & I_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -I_t \quad (4)$$

To derive the pixel displacement (u, v), it is further assumed that the pixels in a certain window have the same motion in the optical flow. Thus, taking an image region with a $w \times w$ window size, there will be totally w^2 equations:

$$\begin{pmatrix} I_x & I_y \end{pmatrix}_k \begin{pmatrix} u \\ v \end{pmatrix} = -I_{tk}, \quad k = 1, \ldots, w^2 \quad (5)$$

Let

$$\mathtt{A} = \begin{pmatrix} (I_x, I_y)_1 \\ \vdots \\ (I_x, I_y)_{w^2} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} (I_{t1}) \\ \vdots \\ (I_{tw^2}) \end{pmatrix} \quad (6)$$

Then the equation can be rewritten as

$$\mathtt{A} \begin{pmatrix} u \\ v \end{pmatrix} = -\mathbf{b} \quad (7)$$

This overdetermined linear system can be solved by minimizing the least squared error, i.e.,

$$\begin{pmatrix} u \\ v \end{pmatrix}^* = -(\mathtt{A}^\top \mathtt{A})^{-1}\mathtt{A}^\top \mathbf{b} \quad (8)$$

Thus, the displacement (u, v) of a pixel can then be calculated accordingly.

The information derived from the optical flow field is the relationship between velocity and depth, not the absolute distance. When a camera is moving forward, the closer regions have larger motion vectors. On the other hand, the smaller motion vectors are associated with the further regions. This fact can be applied to the optical flow for the detection of nearby objects from the image. If the magnitudes of the optical flow vectors are less than a pre-set threshold in an image region, there will be no obstacles along the line-of-sight direction. Although these properties are satisfied when a quadrotor proceeds forward, the optical flow field could be unstable if the motion direction is changed, for example, at the navigation waypoints. In these cases, a false obstacle alarm might be reported. To deal with this problem, one way is to ignore or compensate the optical flow vectors induced by unexpected camera motion or shaking. Since the optical flow field represents different motion directions caused by the movements of the camera and objects in the scene, we can extract the independent motion regions associated with the objects by image segmentation. To verify if a region belongs to an obstacle from the optical flow field, a series of process is conducted.

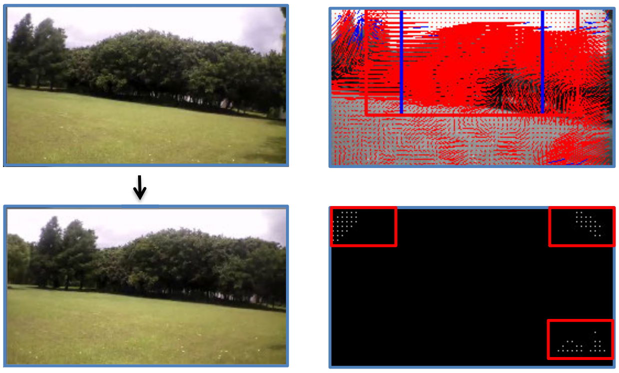


FIGURE 4. The optical flow in the lateral direction due to the control command is compensated using the IMU data.

First, if the motion vectors are larger than a given threshold for over 70% of the optical flow field, it is considered as the result due to the camera shaking and the current image frame is discarded. The onboard IMU data are then used to check the state of the quadrotor. If the lateral motion is instructed by a control command, the optical flow computation is compensated using the IMU data. As shown in Figure 4, there is severe optical flow in the lateral direction. After the motion compensation, only three regions near the corners are preserved, which are associated with the new image content acquired from the scene.

The obstacle detection using the optical flow field is illustrated in Figure 5. Given an input image sequence as shown in Figure 5(a), the optical flow computation result is illustrated in Figure 5(b). The motion vectors marked in blue represent the optical flow with the magnitude larger than a threshold, and the segmentation result is shown in Figure 5(c). In most cases, the obstacles in the foreground will be still mixed with the background. To deal with this problem, we utilize the property that the motion vectors in the image which are associated with the same distance in the scene should possess similar magnitudes. Consequently, the noisy optical flow with uncertainty can be filtered by a clustering process [32]. In the implementation, a sliding window is adopted to combine the motion vectors with comparable magnitudes by the following steps:

- A 5 × 5 sliding window is used to merge the motion vectors with similar magnitudes.

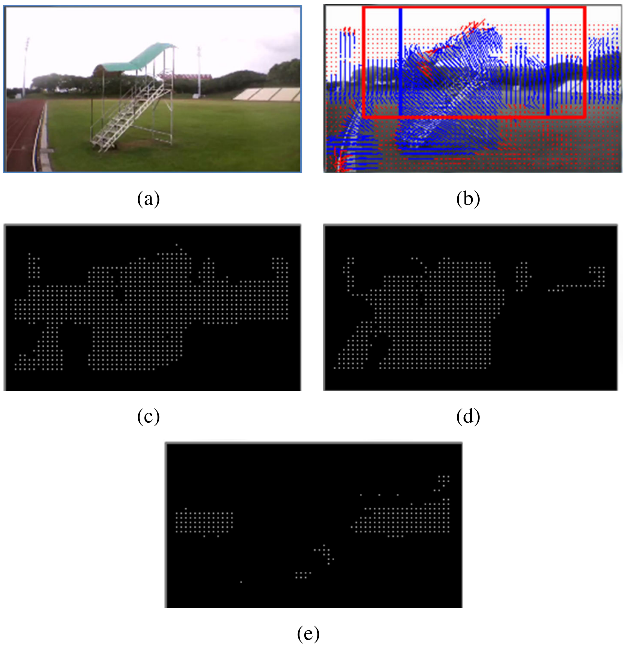


FIGURE 5. **The obstacle detection using the optical flow computation. (a) The original image. (b) The region of interest with motion vectors. Note that the ROI is near the top due to the heading of the quadrotor when proceeds forward. (c) The image before clustering. (d) The foreground region image. (e) The background region image.**

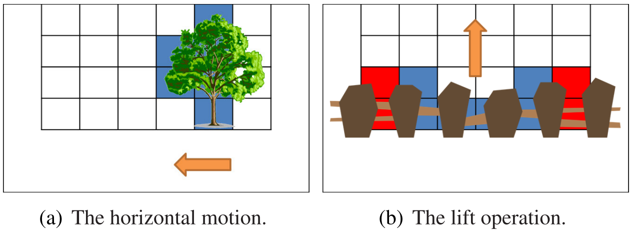

(a) The horizontal motion.
(b) The lift operation.

FIGURE 6. **The safety conditions for robust obstacle avoidance included in the system development.**

- Compare the motion vector not belonging to any clusters with the clustered vector.
- Create a new cluster if the motion vector does not belong to the existing ones.
- Repeat until all motion vectors are clustered.

By removing the outlier motion vectors with extremely small or large values, the foreground and background are shown in Figures 5(d) and 5(e), respectively. It should be noted that the optical flow computation is used for online analysis in the real scene. There is no ground truth data for evaluation in the implementation.

To make the obstacle avoidance algorithm more robust in terms of definite threats under various situations, two safety conditions are included in the system development as illustrated in Figure 6. First, if the object area is not sufficiently large in the RoI (more than 50% in the experiments), it is assumed the obstacle is beyond the restricted range and the quadrotor proceeds as planned. This setting is to ensure that the quadrotor detours when the obstacle is sufficiently close and may have a potential threat. Second, the motion direction of the flight will be changed only if the obstacles are detected in several consecutive image frames. This criterion is adopted to avoid the false alarms by temporal filtering. As for the strategy for obstacle avoidance, the quadrotor will turn to the opposite side of the obstacle until it is out of the RoI. These include the left and right turns, and the lift operation. More advanced maneuver might be investigated by consider the power consumption and other motion control issues.

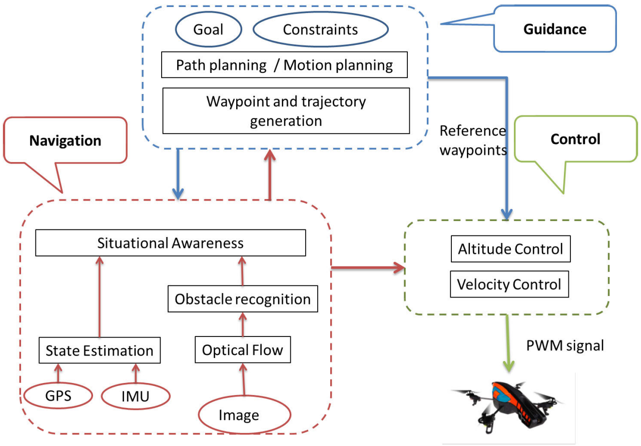


FIGURE 7. **The system flowchart of our autonomous navigation technique. It consists of three subsystems for the autonomous flight: (1) guidance system, (2) navigation system, and (3) control system.**

III. AUTONOMOUS NAVIGATION SYSTEM

The obstacle avoidance algorithm presented in the previous section is integrated with the path planning technique to build an autonomous navigation system. As illustrated in Figure 7, it consists of three subsystems for the autonomous flight: (1) guidance system, (2) navigation system, and (3) control system [33]. The path planning is used for flight guidance, autonomous navigation, and motion control. It can be divided into two parts, the static and dynamic planning. The objective of static planning is to derive an optimal navigation path in advance based on the map information. During the autonomous navigation, it is required to adjust the planned motion path if obstacles are encountered. The dynamic planning is to integrate the obstacle avoidance algorithm and generate a new path to the destination.

The static path planning is carried out in the offline environment, with the consideration of terrain restriction and quadrotor limitation. It provides the destination, waypoints, and other navigation constraints such as the flight altitude. If there are no obstacles in the environment, the ideal planned path is a straight line between two waypoints. The objective is to provide a safe trajectory for quadrotor navigation. To fulfill this purpose, the large objects such as buildings and man-made structures are modeled in the virtual environment for path planning. In the implementation, we use the satellite images obtained from Google Earth for 3D modeling of the environment, as illustrated in Figure 8 (the generated

FIGURE 8. The satellite images obtained from google earth for 3D modeling of the environment.

3D model is shown in Figure 13). The GPS coordinates are used for the building setting. The RRT algorithm is developed for path planning with the knowledge of the working space [34]. The generated 3D waypoints are then transferred to GPS coordinates for the real scene navigation.

In the early research, Kavraki *et al.* proposed a probabilistic roadmap approach for path planning [35]. They presented a two-phase method to deal with the robot path planning problem in static workspaces. The rapidly-exploring random tree (RRT) algorithm is first presented by LaValle *et al.* [36], and followed by further modifications and applications [37], [38]. It is originally designed for a broad class of path planning problems, and is suitable for complex and high dimensional spaces. However, the motion cost is not explicitly considered in the traditional RRT approach. In this work, we use the RRT* algorithm for optimal path planning [39]. It is a sampling-based planning algorithm which is able to quickly find an initial path and perform an optimization for the path. We first use the map from Google Earth to mark the obstacle location, and then build the three-dimensional environment. The pseudo code of our RRT* implementation is described in Algorithm 1.

Algorithm 1 Probabilistically Optimal RRT

```
$\tau \leftarrow$ Initial Tree;
$\tau \leftarrow$ Insert Node;
for  $i = 1$  to  $i = N$  do
    $Z_{rand} \leftarrow$ Sample
    $Z_{nearest} \leftarrow$ Nearest Point
    $(Z_{new} U_{new}) \leftarrow$ Steer
    if ObstacleFree $Z_{new}$, then
        $Z_{near} \leftarrow$ Near
        $z_{min} \leftarrow$ ChooseParent
        $\tau \leftarrow$ InsertNode
        $\tau \leftarrow$ Rewire
    end if
end for
return $\tau$
```

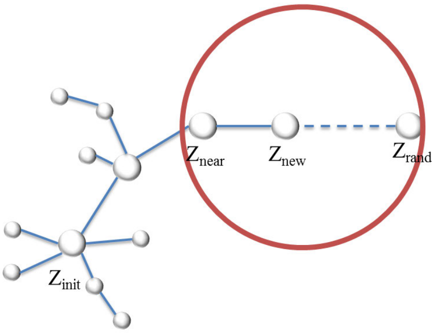


FIGURE 9. The sampling process of the RRT* algorithm.

The RRT* algorithm is extended from the traditional RRT algorithm with an additional cost function. It takes the neighborhood of a new point into consideration to compute the cost of the entire path. As illustrated in Figure 9, the path is updated at each iteration according to the cost of the path from Z_{init} to Z_{new}. Thus, the path planning process output is a series of waypoints for quadrotor navigation. When the quadrotor reaches a waypoint, the next waypoint will be used to guide it one step towards the destination. In the pseudo codes, RRT* starts from the root of the tree. For each iteration, the tree is built with random sample points from the space. The nearest routine will find the node in the tree nearest to the new random sample point. The steering routine generates a new point reachable to the random point. It is then verified if the nearest point satisfies the environment constraint. The new point will be added to the tree if its path to the nearest point is collision free. If the distance between the current position and the goal is smaller than a threshold, the procedure is terminated and an initial path is derived. RRT* will continue updating the path while the number of the iterations increases. In the experiments, it is implemented for off-line processing to provide sufficient iterations. The path planning result is then integrated with the 3D simulation environment constructed according to the real scenes using OpenGL.

During the flight navigation, the quadrotor follows the path provided by the initial planning. When the quadrotor moves forward, the navigation system will use the onboard camera to detect the approaching obstacles by the optical flow based avoidance technique described in the previous section. If the quadrotor detours from the initial path according to the commands sent by the obstacle avoidance system, the guidance system will generate a new path based on the quadrotor position and the next waypoint. For the autonomous flight system, we propose two control strategies, namely following the planned path and the obstacle avoidance scheme. As shown in Figure 10, the control system receives the commands from the obstacle avoidance system, and transmit the PWM signals to each motor to adjust the pose of the quadrotor. Based on the optical flow field, the quadrotor will be instructed to turn away from the large motion vectors.

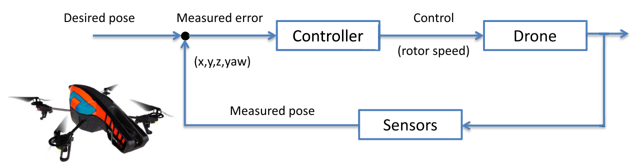


FIGURE 10. The control scheme of the quadrotor UAV for the dynamic path planning.

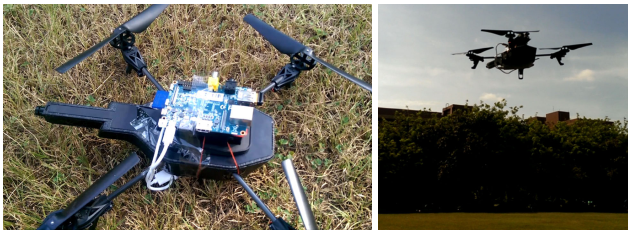


FIGURE 11. The parrot AR. Drone with our attached single-board computer for online processing.

TABLE 1. The specification of the commercially available quadrotor, AR. Drone, used in the experiments.

Type	Specification
Camera	1280 × 720
Processor	32-bit ARM Cortex A8 1 GHz
RAM	1 GB DDR2 RAM at 200 MHz
Sensors	accelerometer, gyroscope, pressure sensor
Power	14.5W / 28,500 rpm
Battery	12 min flight time
Wireless	Wi-Fi b/g/n
BVLOS	50.3 m

IV. EXPERIMENTS AND EVALUATION

To evaluate the proposed vision guided quadrotor navigation techniques, several experiments are conducted with low altitude flight. The autonomous navigation is carried out in a few outdoor environments containing obstacles. Figure 11 shows the UAV system used in our experiments. It is developed based on a commercially available quadrotor, Parrot AR. Drone, with hardware and software modifications. The specification of the quadrotor is tabulated in Table 1. An embedded single-board computer attached on the quadrotor is used as our control platform. It is an ARM based system running on the Linux OS. The image sequences are acquired from the built-in camera of the quadrotor, and transmitted to the onboard embedded system via the Wi-Fi communication for processing. The control commands are then sent back to the quadrotor.

In our experiments, the image processing time for a frame on the embedded platform is about 150-180 *ms*, which corresponds to about 6 fps (frames per second). The optical flow and other image processing algorithms are implemented based on the OpenCV library with slight modification and parameter settings for the application scenarios. In the experiments, the normal UAV flying speed is about 1 *m/s*. There will be about 6 frames under processing for 1 meter fly. The UAV will slow down when it is close to an obstacle for direction change and avoidance. Consequently, the UAV flying speed does not affect the image processing computation in a noticeable way.

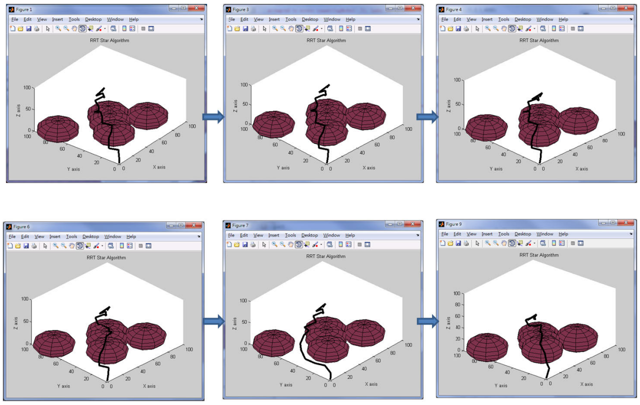


FIGURE 12. The path planning result from RRT*.

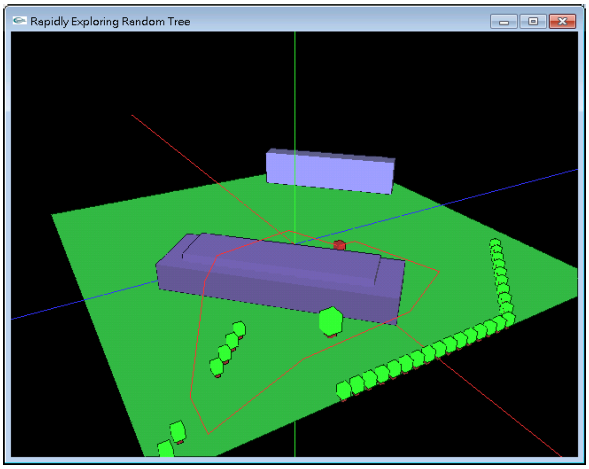
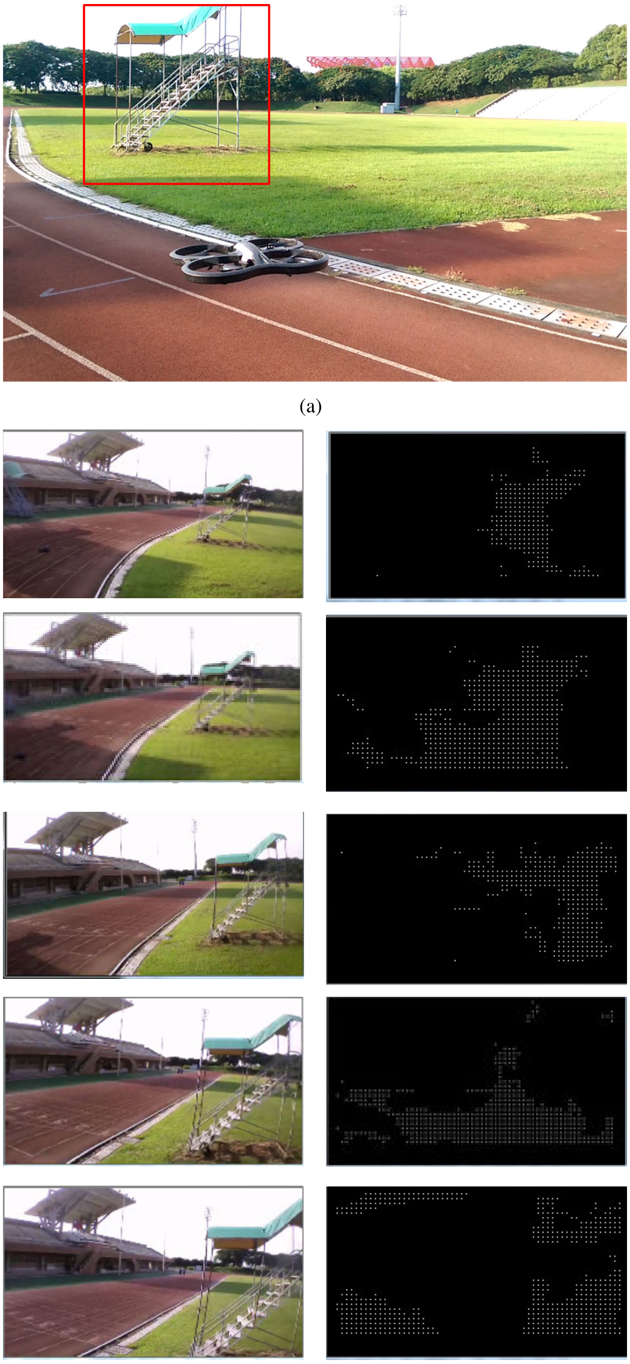


FIGURE 13. The 3D simulation environment used for path planning experiments. There are two workable paths (marked in red lines) in this figure.

Before the quadrotor's navigation in the outdoor environment, the path planning is carried out first to provide a safe route. The RRT* algorithm is used to generate an initial path. Figure 12 shows the RRT* simulation results with the cost of the entire navigation path. The optimal path is updated based on the generated random tree for each iteration. In the experiment, the numbers of iterations are set as 500 and 800 for comparison. The random tree spans wider when the number of iterations increases, and it is more likely to obtain the optimal path. Figure 13 shows the RRT* path planning result visualized using OpenGL for an outdoor environment. The two workable paths marked in red are both near optimal and collision-free. If there is no restriction on the flight altitude, it is always possible to raise the quadrotor high enough and proceed in a straight line. Nevertheless, we are more interested in the autonomous flight in low altitude environments.



(a)

(b)

FIGURE 14. (a) The iron stair as an obstacle is indicated by the red box. (b) A sequence of images captured from the onboard camera during the flight.

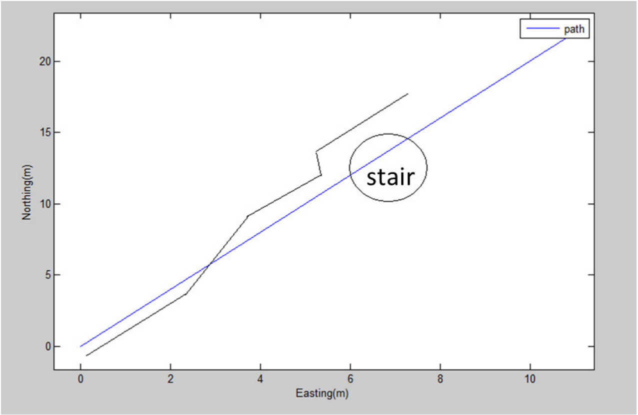


FIGURE 15. The trajectory of an outdoor experiment. The GPS plot shows that the quadrotor can avoid an iron stair while moving forward.

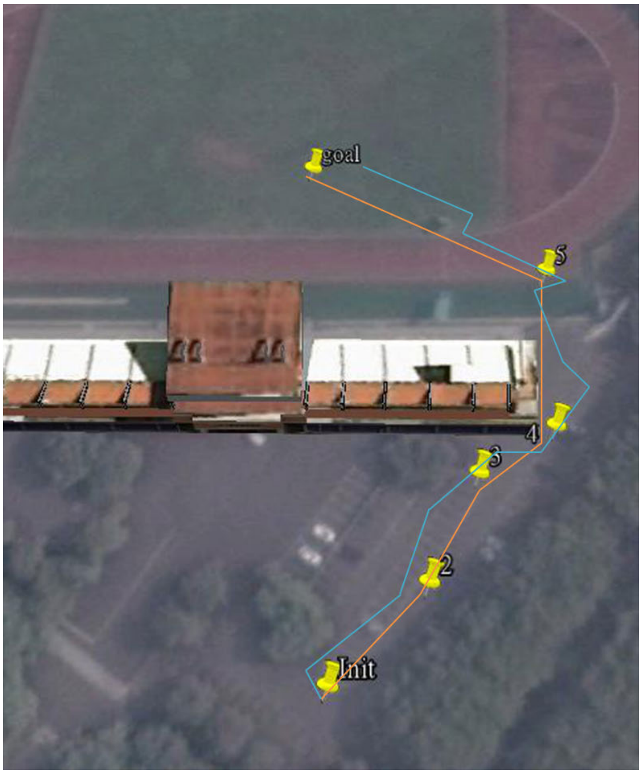


FIGURE 16. The initial path as planned (yellow line) and the actual path recorded by the quadrotor (blue line). The obstacle is bypassed during the flight.

In the first experiment, the quadrotor is moving towards an iron stair as shown in Figure 14(a). Several captured images and the optical flow field are shown in Figure 14(b). It is to verify if our method can detect and avoid the obstacle in the waypoint direction. When the quadrotor moves along the planned path, the system will check whether it satisfies the warning condition. If the criteria are met, the flight direction will be adjusted by the control system for collision avoidance. Figure 15 shows the trajectory bypassing the iron stair. The guidance system will redirect the path toward the next waypoint after the obstacle is disappeared in the image.

We have also performed the obstacle avoidance for various situations, including fence, trees, and street light. The results for several tests and success rates are tabulated in Table 2. It shows that our algorithm performs well in the single obstacle cases such as the tree or iron stair. The low success rate for

TABLE 2. The experiment results in the outdoor environment.

Type	Obstacles	Tests	Success Rate
Tree	1	10	100
Iron stair	1	12	100
Fence	1	7	43
Trees	3	7	43
Street light	1	6	66

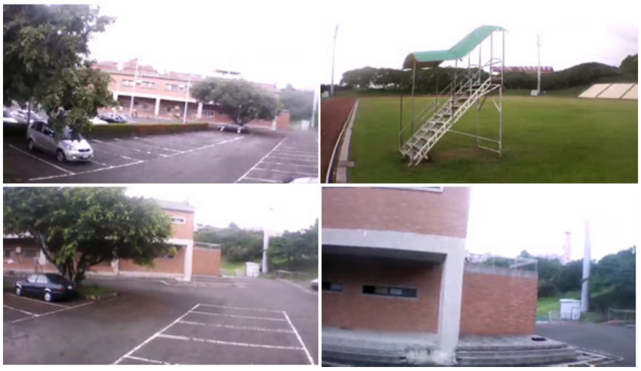


FIGURE 17. The images captured when the obstacles are encountered.

multiple obstacles is mainly due to the optical flow field not able to separate the objects and the background very clearly.

Finally, the proposed vision guided navigation with path planning is tested in a large scale environment. Figure 16 shows the experiment carried out in a football field of about 200 × 200 square meters containing a large building. The planned path is shown in orange lines on the Google map, and the waypoints are indicated by the yellow thumbtacks. The blue lines in the figure show the trajectory of the actual flight. It consists of several flight direction changes due to the encountered obstacles (tree, stair and building) as shown in Figure 17. In addition, the quadrotor makes turns earlier as planned at the thumbtack location. This is due to a preset safety distance of 5 meters to compensate the limited GPS accuracy.

V. CONCLUSION

In this work, we present a vision guided autonomous flight system for quadrotor navigation. It consists of path planning and real-time obstacles avoidance techniques. Our method can detect the frontal obstacles and guide the quadrotor with a proper direction. The proposed algorithms are developed for an embedded platform to achieve the online processing capability. We have reduced the computational requirement of the optical flow method. The experiments are carried out in the outdoor environment with single or multiple obstacles. The results demonstrate the feasibility of our approach for vision guided quadrotor navigation. In the current system, we do not specifically consider the presence of moving obstacles. More advanced image-based approaches will be investigated in the future work.

REFERENCES

[1] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[2] C. Stöcker, R. Bennett, F. Nex, M. Gerke, and J. Zevenbergen, "Review of the current state of UAV regulations," *Remote Sens.*, vol. 9, no. 5, p. 459, May 2017.

[3] P.-C. Chen, Y.-C. Chiang, and P.-Y. Weng, "Imaging using unmanned aerial vehicles for agriculture land use classification," *Agriculture*, vol. 10, no. 9, p. 416, Sep. 2020.

[4] J. Kwak and Y. Sung, "Autonomous UAV flight control for GPS-based navigation," *IEEE Access*, vol. 6, pp. 37947–37955, 2018.

[5] S. Bouabdallah and R. A. Siegwart, "Full control of a quadrotor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 153–158.

[6] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. Robot. Syst.*, vol. 57, no. 1, pp. 65–100, Jan. 2010.

[7] H. Liu, X. Wang, and Y. Zhong, "Quaternion-based robust attitude control for uncertain robotic quadrotors," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 406–415, Apr. 2015.

[8] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, "A survey of quadrotor unmanned aerial vehicles," in *Proc. IEEE Southeastcon*, Mar. 2012, pp. 1–6.

[9] J. Su, D. Yi, B. Su, Z. Mi, C. Liu, X. Hu, X. Xu, L. Guo, and W.-H. Chen, "Aerial visual perception in smart farming: Field study of wheat yellow rust monitoring," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2242–2249, Mar. 2021.

[10] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[11] Y. Lu, Z. Xue, G. S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-Spatial Inf. Sci.*, vol. 21, no. 2, pp. 21–32, 2018.

[12] X. Peng, H. Lin, and J. Dai, "Path planning and obstacle avoidance for vision guided quadrotor uav navigation," in *Proc. 12th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2016, pp. 984–989.

[13] W. Yang, J. Tang, R. He, and Y. Chen, "A medium-term conflict detection and resolution method for open low-altitude city airspace based on temporally and spatially integrated strategies," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 1817–1830, Sep. 2020.

[14] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *J. Intell. Robot. Syst.*, vol. 87, no. 1, pp. 141–168, Jul. 2017.

[15] A. Al-Kaff, Q. Meng, D. Martin, A. de la Escalera, and J. M. Armingol, "Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 92–97.

[16] X. Yang, J. Chen, Y. Dang, H. Luo, Y. Tang, C. Liao, P. Chen, and T. Cheng, "Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 156–167, Jan. 2021.

[17] L. He, N. Aouf, J. F. Whidborne, and B. Song, "Integrated moment-based LGMD and deep reinforcement learning for UAV obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 7491–7497.

[18] J.-P. Sanchez-Rodriguez and A. Aceves-Lopez, "A survey on stereo vision-based autonomous navigation for multi-rotor MUAVs," *Robotica*, vol. 36, no. 8, pp. 1225–1243, Aug. 2018.

[19] J. Park and Y. Kim, "Collision avoidance for quadrotor using stereo vision depth maps," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 3226–3241, Oct. 2015.

[20] H. Oleynikova, D. Honegger, and M. Pollefeys, "Reactive avoidance using embedded stereo vision for MAV flight," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 50–56.

[21] S. Tijmons, G. C. de Croon, B. D. Remes, C. De Wagter, and M. Mulder, "Obstacle avoidance strategy using onboard stereo vision on a flapping wing MAV," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 858–874, Aug. 2017.

[22] J. Byrne, M. Cosgrove, and R. Mehra, "Stereo based obstacle detection for an unmanned air vehicle," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2006, pp. 2830–2835.

[23] G. Cho, J. Kim, and H. Oh, "Vision-based obstacle avoidance strategies for MAVs using optical flows in 3-D textured environments," *Sensors*, vol. 19, no. 11, p. 2523, 2019.

[24] K. McGuire, G. de Croon, C. D. Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1070–1076, Apr. 2017.

[25] N. Urieva, J. McDonald, T. Uryeva, A. S. R. Ramos, and S. Bhandari, "Collision detection and avoidance using optical flow for multicopter UAVs," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 607–614.

[26] J. Chen, Y. Zhou, Q. Lv, K. K. Deveerasetty, and H. U. Dike, "A review of autonomous obstacle avoidance technology for multi-rotor UAVs," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Aug. 2018, pp. 244–249.

[27] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2443–2451.

[28] R. Timofte and L. Van Gool, "Sparse flow: Sparse matching for small to large displacement optical flow," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 1100–1106.

[29] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proc. 13th Scandin. Conf. Image Anal.* Berlin, Germany: Springer-Verlag, 2003, pp. 363–370.

[30] K. P. B. Horn and G. B. Schunck, "Determining optical flow," Cambridge, MA, USA, Tech. Rep., 1980.

[31] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[32] M. Narayana, A. Hanson, and E. Learned-Miller, "Coherent motion segmentation in moving camera videos using optical flow orientations," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1577–1584.

[33] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315–378, 2012.

[34] J. Kim and J. P. Ostrowski, "Motion planning a aerial robot using rapidly-exploring random trees with dynamic constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Sep. 2003, pp. 2200–2205.

[35] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[36] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[37] J. J. Kuffner, Jr. and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Apr. 2000, pp. 995–1001.

[38] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[39] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

HUEI-YUNG LIN (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the State University of New York at Stony Brook, USA. In 2002, he joined the Department of Electrical Engineering, National Chung Cheng University, Taiwan, as an Assistant Professor, where he is currently a Full Professor. He has served as the Director for the Research Liaison Division, from 2009 to 2013, and the Academic Development Division, from 2012 to 2014, both with the Office of Research and Development. He is also a Professor and the Director of the Robot Vision Laboratory, National Chung Cheng University. He has authored more than 170 journal articles and conference papers, and has written two book chapters. He also holds 11 U.S. patents and nine Taiwan patents. His research interests include computer vision, robotics, machine learning, and image processing. He is a Senior Member of OSA. He serves as an organizing committee member and a program committee member for more than 50 international conferences. He was a recipient of the Outstanding Robotics Engineer Award from the Robotics Society of Taiwan and the Outstanding Academic-Industry Cooperation Award from the Taiwan Association of System and Science and Engineering.

XIN-ZHONG PENG received the B.S. degree in electronics engineering from Chung Yuan Christian University, and the M.S. degree in electrical engineering from the National Chung Cheng University, Taiwan. He is currently with Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan. His research interests include robotics, embedded systems, mechatronics, and computer vision.