2019 International Conference on Unmanned Aircraft Systems (ICUAS)
Atlanta, GA, USA, June 11-14, 2019

A Vision-based Unmanned Aircraft System for Autonomous Grasp & Transport*

Xu Liu[1], Yuqing He[2], Bo Chen[1], Yongqiang Hou[3], Kaiyuan Bi[4], Decai Li[5]

Abstract— **The progress in sensor technologies, computer capabilities and artificial intelligence has endowed the unmanned aircraft system (UAS) with more autonomous abilities. Motivated by the 6th International Unmanned Aerial Vehicle Innovation Grand Prix (UAVGP), a UAS with high degree of autonomy was developed to perform the mission of building a simulated tower using prefabricated components. According to the requirement of the competition, the UAS was designed and implemented from the following four parts: 1) navigation and control, 2) recognition and location, 3) grasp and construction, and 4) task planning and scheduling. Different levels of autonomy have been given to the UAS based on these parts. The system hardware was developed on a quadrotor platform by integrating various components, including sensors, computers, power and grasp mechanism. Software which included precise navigation, mission planning, real-time perception and control was implemented and integrated with the developed UAS hardware. The performance in the test environment and actual competition showed that the UAS could perform the mission without human intervention with high autonomy and reliability. This paper addresses the major components and development process of the UAS and describes its application to the practical mission.**

I. Introduction

Over the past several decades, unmanned vehicle systems have been considerably improved thanks to the technological advances in sensor, data processing, and machine learning. Among various types of unmanned vehicle system, unmanned aerial vehicles (UAVs) or unmanned aircraft systems (UASs) have attached a great deal of interest for their capacity to execute dangerous and difficult missions ranging from military to civil application in the air environment [1], [2]. However, most existing UAVs are still remotely controlled by human operators with continuous intervention, which implies that they are not “unmanned”, let alone autonomous. To make unmanned vehicles truly “unmanned”, a number of efforts and researches have been made [3], [4].

In line with the research trend, the Unmanned Aerial Vehicle Innovation Grand Prix (UAVGP) was held on November 2-4, 2018, in Anji, China. Fig. 1(a) shows the competition course layout. In brief, the contestants were required to design a unmanned rotorcraft to autonomously transport building component (which was simulated as a box in the competition, see Fig. 1(b)) from one platform (which was called component repository) to the other (which was called building site), and stack them up as high as possible. To complete the mission, the UAS was required to have a multilayered autonomy. Respectively, low level of autonomy required the UAS to have the basic abilities of autonomous take-off and landing depending on its control and navigation system. Middle level of autonomy required the assistance of external perceptual sensors and mechanical structure to perform more complex operations, e.g. positioning and grasp. High level of autonomy included task and route planning, which provided top-level commands for the whole system. The three supplemented and interacted with each other.

The team from Shenyang Institute of Automation, Chinese Academy of Sciences participated in the competition and developed a sophisticated UAS based on a quadrotor platform. Various hardware modules included navigation, propulsion, perception and grasp were designed. Accordingly, the associated codes implemented by C++/python in three different computers were integrated with the developed hardware, including precise navigation, mission planning, real-time perception and control. After a series of hardware adjustments and software updates, test experiments and actual competition showed that the developed UAS had reliable, efficient performance.

The remainder of the paper is organized as follows. Section Ⅱ and Section Ⅲ demonstrate the major components of the UAS in the hardware and software respectively, and some strategies applied to the mission were described. The performance in actual environment was shown in Section IV. Finally, the conclusion of the study was illustrated in Section Ⅴ.

II. UAS Hardware Development

Fig. 2 shows the developed UAS which was based on a quadrotor and its hardware architecture is described in Fig. 3. A quadrotor was selected instead of a helicopter mainly because of its maneuverability [5]. The hardware system of

*This study was supported by the National Natural Science Foundation of China (Nos. U1508208 and U1608253) and the Science and Technology Planning Project of Guangdong Province (2017B010116002).

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016 China; Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China; University of Chinese Academy of Sciences, Beijing 100049, China (email: *liuxu1@sia.cn*(corresponding author); *chenbo@sia.cn*)

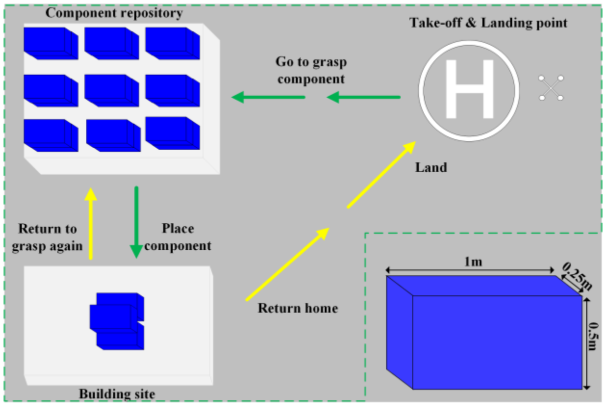
[2]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China; Shenyang Institute of Automation (Guangzhou), Chinese Academy of Sciences, Guangzhou 511458, China (email: *heyuqing@sia.cn*)

[3]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; College of Information Science and Engineering, Northeastern University, Shenyang 110016, China (email: *houyongqiang@sia.cn*)

[4]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; Information and Control Engineering Faculty, Shenyang Jianzhu University, Shenyang 110016, China (email: *15174175810@163.com*)

[5]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China (*lidecai@sia.cn*)

978-1-7281-0332-7/19/$31.00 ©2019 IEEE



(a) (b)

Fig. 1 (a) Schematic diagram of competition field (about 50m * 40m). (b) The building component: 1m * 0.25m * 0.5m.

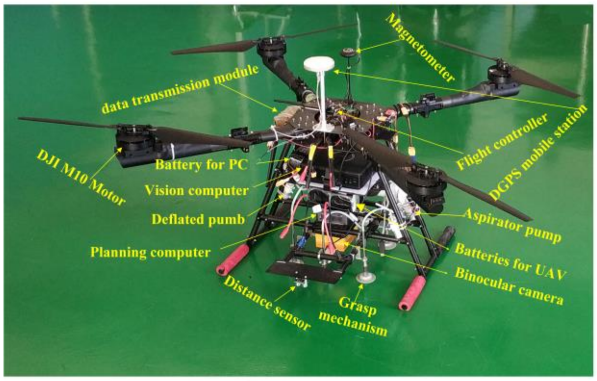


Fig. 2 The developed USV.

the developed UAS comprised four subsystems: 1) flight system, 2) perception system, 3) computer system, and 4) grasp system. Details of these subsystems are described below.

A. Flight System

The flight system provided essential propulsion, control, and navigation module. Four identical motors whose maximum pull was 14kg per axis were installed on the airframe. For a long endurance, two 6S Li-Po batteries, 16000mAh each, were mounted on the UAS, which made the endurance of the UAS reached 11 minutes with an eight-kilogram load. An inertial measurement unit (IMU) and difference GPS (DGPS) was used to provide vehicle motion information in the high-frequency and low-frequency range, respectively. The complementary navigation module by combining IMU and DGPS, along with an magnetometer, would yield accurate navigation information, which enabled the UAS to successfully perform the mission. TABLE I shows its major characteristics.

B. Perception System

The perception system obtained the three-dimension position of the components and determined current state of the UAS by interacting with the environment. In addition to the navigation sensors, the perception system included two parts: vision module and ranging module. The vision module was used to acquire the accurate position of the components, which consisted of a downward-looking binocular camera and an onboard computer. The connection between the binocular camera and the airframe was rigid to keep the relative position between them constant, which was required by subsequent coordinate transformation. The ranging module included a distance sensor that was also installed downward to measure the distance between the UAS's bottom and the object. The measurements were used later as criteria to judge whether the components have been grasped or placed successfully.

TABLE I

MAJOR CHARACTERISTICS OF THE DESIGNED UAS

Characteristic	Value
Diagonal wheelbase	1.9m
Maximum Takeoff Weight	25kg
Dry Weight	19kg
Maximum Flight Duration	11min
Dimensions (folded)	0.95m*0.95m*0.7m (0.65m*063m*0.7m)

C. Computer System

A computer system that comprised two onboard computers and an external one was used to handle the vision measurements, issue commands, and conduct real-time display. The vision computer was required to have sufficient computational capacity to process the vision measurement data acquired by the binocular camera in a frame rate of 10 fps. Another computer onboard was used for task planning and scheduling, i.e. acquired the current status of the UAS then perform an appropriate action depended on it. The two onboard computers could work reliably under various environment disturbance due to vibration impact, heat, and operational uncertainty. The external computer was used to run the ground station (GS) programs, whose function was to display important real-time information of the UAS and execute some essential operations. Based on the above considerations, the computer system was designed to consist of three separate computers (ZOTAC, Intel NUC, and ThinkPad T470). These three computers along with the flight controller were connected via serial communication link used Mavlink communication protocol.

D. Grasp System

Fig. 4 demonstrates the grasp system used to grasp the components. Considering the size, shape and surface of the components, the traditional grasp devices, e.g. tongs and manipulators, were not competent, so a novel mechanism was designed and implemented. Six suckers in total were divided into three groups, and each of them could work separately to increase the success rate of grasping. Two different kinds of pump were used to grasp and release the components, which was controlled by a servo depending on the current state of the UAS.

Due to the precision of hardware devices and existence of stochastic errors, the navigation deviation could still reach ten centimeters even used differential GPS and tuned the parameters to almost optimal. To compensate for the positioning error, a scalable device whose length could vary

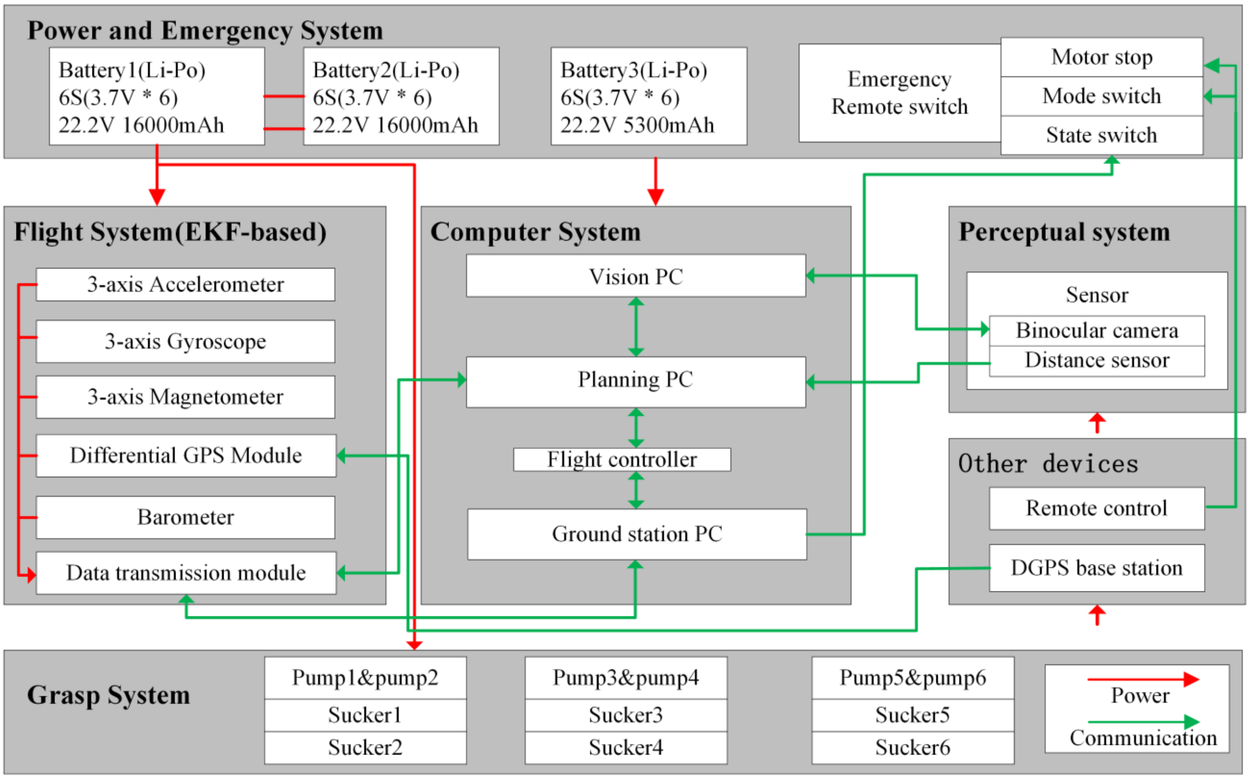


Fig. 3 The hardware structure of the designed UAS.

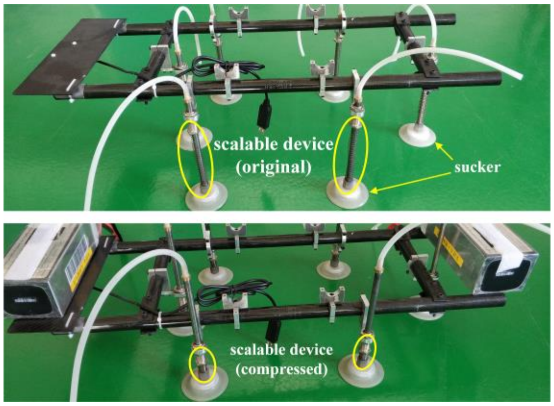


Fig. 4 The grasping system.

from 5cm to 15cm according to the magnitude of the contact force was applied (shown in the yellow oval in Fig. 5). Once the UAS touched the component and continued to descend, the device gradually shrinks from natural length, which ensured both the stability of the UAS and the success rate of grasping.

III. SOFTWARE STRUCTURE

The software structure of the developed UAS is graphically demonstrated in Fig. 5. As shown in the figure, the software structure could be divided into four main components: 1) The programs ran in the flight controller, including guidance, navigation, and control, which provided the lowest level of autonomy. 2) The vision programs played the role of perception, which was used to get the relative position of the component with respect to the camera. These provided the intermediate level of autonomy. 3) The task planning and scheduling programs directed the whole mission by defining the corresponding action for each state. It provided the highest level of autonomy. 4) Fault-tolerant measures that included emergency processing and relevant strategies to handle the emergent failure and improve the practical performance. It provided assisted autonomy.

A. Navigation and Control

The navigation and control algorithms of a UAS commonly comprise two part: attitude estimation and attitude control. In the following these two parts will be presented according to the order.

Attitude estimation was to merge measurements from different sensors, including a gyro, a accelerometer, and an magnetometer to figure out the attitude of the UAS. Generally, there are three main algorithms to calculate the attitude of the UAS: 1) Explicit Complement Filter [6], [7]. 2) Gradient Descent [8], [9], and 3) Extended Kalman Filter (EKF). Considering the UAS would have a sudden acceleration and deceleration frequently, and the attitude fluctuated greatly because of the wind and varying load, the last one EKF was selected because of its better dynamic performance, although it has greater computation complexity. Dual-subsample rotation vector method was used to calculate the optimal estimation of attitude from the measurement data, which turned out to be robust and accurate even in the case of dynamic condition. With setting appropriate parameter of EKF, the accuracy can reach 1°for attitude, and 2cm for position according to the actual flight data. In addition, this combination was also used for velocity and position prediction, in which the DGPS provides position observations that used for state correction.

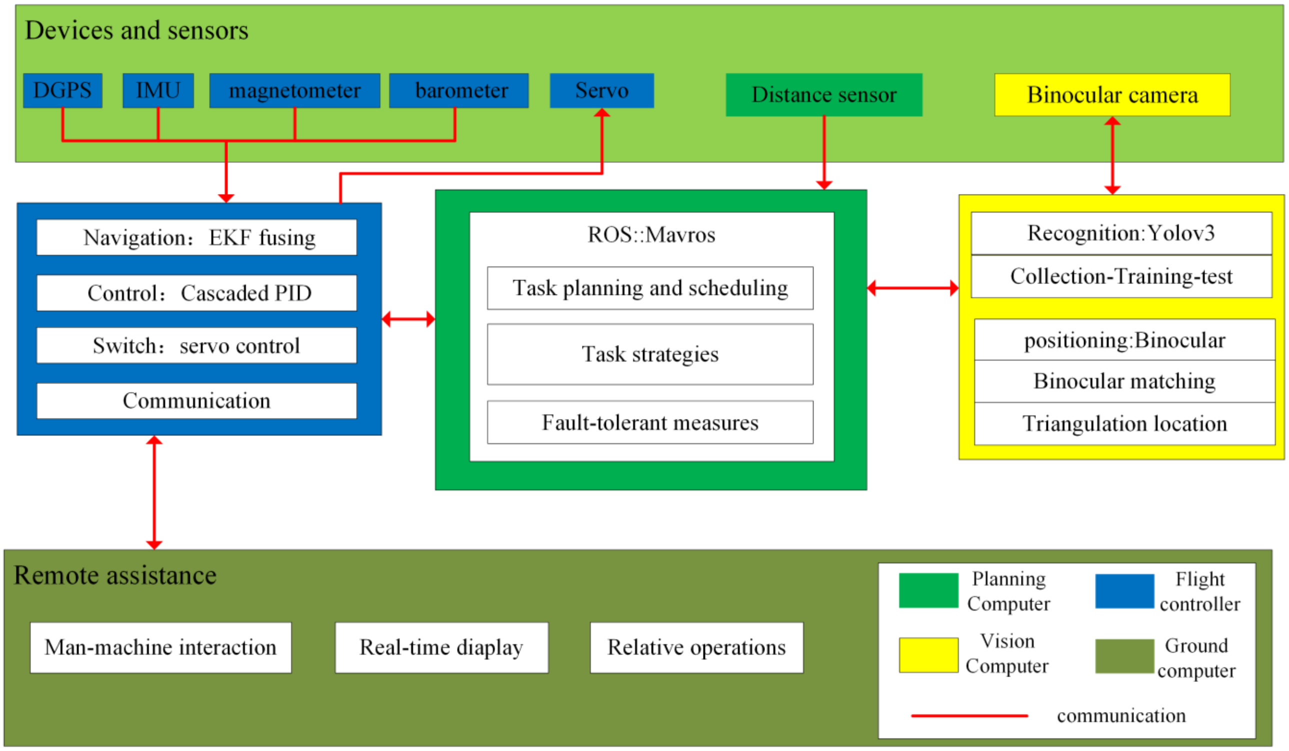


Fig. 5 A schematic diagram of the software structure.

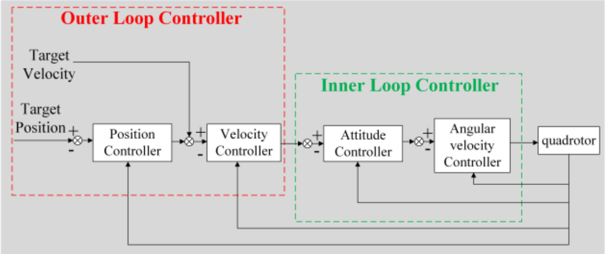


Fig. 6 The cascaded PID controller structure for the designed UAV.

The quadrotor is a typical underactuated system. The following state vector can describe the UAS's motion:

$$\overline{x} = [x\ y\ z\ v_x\ v_y\ v_z\ \theta\ \phi\ \psi\ \dot{\theta}\ \dot{\phi}\ \dot{\psi}] \qquad (1)$$

Where, x, y, and z represent the position of the UAS in the local north-east-down(NED) coordinate system. v_x, v_y, and v_z are the UAS's velocity in the same coordinate system. θ, ϕ, and ψ are the orientation in Euler angles, while $\dot{\theta}$, $\dot{\phi}$ and $\dot{\psi}$ are angular velocity. The motion model in (1) represents 6-DOF motion of the UAS. Considering from the perspective of practicality, the conventional proportional-integral-derivative (PID) controller was applied to determine the required output of the electric motor. In the designed UAS, the PID controller was a hierarchical architecture comprised the position and velocity controller as the outer loop and the attitude and angular velocity controller as the inner loop. Compared with the single-loop PID controller, the cascaded PID controller was more stable for the existence of angular velocity inner loop. Fig. 6 describes the control structure of the flight controller.

B. *Recognition and Location*

In the competition, the components' position coordinate was not given in advance, so the vision programs were implemented to perceive the surrounding environment and acquired the position of the components.

The existing algorithm for target recognition can roughly divided into two categories: one is the traditional [10]–[14], the other is deep learning based [15]–[17]. Compared to the traditional one, the deep learning based have a relative high identification precision but with lower speed. For the given mission of grasping a component, compared to the speed, the precision was given priority. Considered comprehensively from speed and accuracy, the Yolo3 algorithm was selected to perform the recognition mission [18].

To achieve the goal of identifying the components successfully, a three-step procedure was executed. First, collected and labeled training samples. The complexity and richness of the sample should be adequate to adapt to the varying lighting condition in outdoor environment and recognition confusion resulted from the dense layout of the components. The process was implemented by a manually operated drone with a binocular camera mounted on it. Second, trained the model. To make the model valid for the given target, it was required to be trained by the labeled samples to acquire the expected parameters int the vision computer. During the training process, the component was the only one class of object. Last, tested the model through practical flight. If the actual performance of the model, especially the recognition accuracy was unsatisfied, the model was required to be retrained by increasing or replacing the training samples.

A downward-looking binocular camera was mounted on the UAS to locate the components. Compared to the

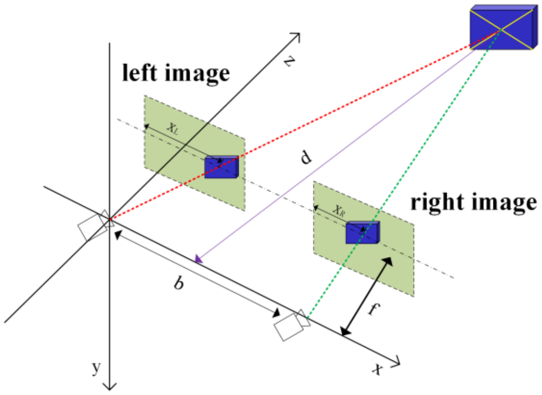


Fig. 7 Schematic diagram of binocular ranging.

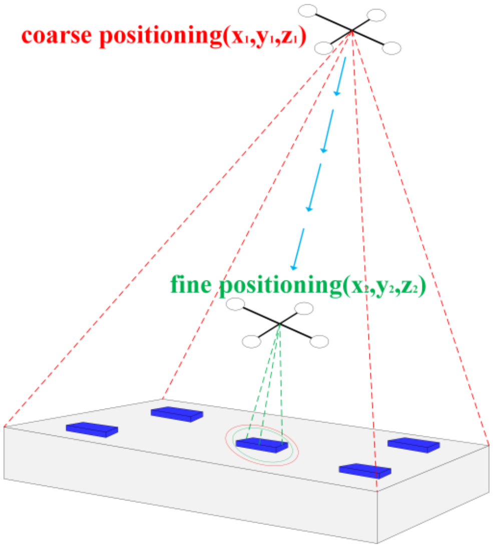


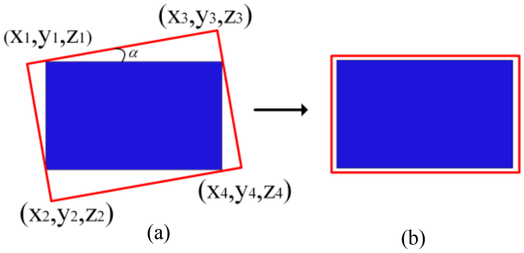
Fig. 8 The positioning process.

monocular visual orientation and structured light ranging, the binocular location system has larger Field of View (FOV) that can provide more depth information and higher positioning accuracy and can work under strong light in the outdoor environment. Fig. 7 demonstrates the principle of binocular ranging, which is based on the binocular matching to figure out the parallax, then uses the principle of similar triangle to calculate the depth of the target point, and the depth d can be calculated as following:

$$d = \frac{b \cdot f}{X_R - X_T} = \frac{b \cdot f}{x} \tag{2}$$

Here, b was the binocular distance, f was the camera focal length, X_R , X_T was the camera coordinate of same point in two images. Next, the coordinate information of the target point under the camera coordinate system could be simply acquired, while the guidance coordinate system used by the UAS is the local NED, so a coordinate transformation between the two coordinates was required, which could be presented by the following formula:

$$P_n = RP_c + b + P_{nc} \tag{3}$$



(a)

(b)

Fig. 9 The process of yaw adjustment.

Where, R denoted the 3×3 transfer matrix, P_n , P_c was the local NED coordinate and camera coordinate of the object, P_{nc} was the current local NED coordinate measured by DGPS, vector b denoted the bias between the center of the left camera and the bottom of the airframe. In practical application, the incremental average was used to reduce the error in the transformed coordinates resulted from the airframe vibration and random error. Until now, the three-dimension coordinate information of the component in the local NED coordinate could successfully acquire.

C. Grasp and Construction

In order to obtain larger FOV that had the potential of containing more components, the UAS was required to be further away from the component repository, which would result in lower location accuracy because of the longer distance. The contradiction between the size of the perceived region and the positioning precision generated the hierarchical positioning strategy. First, the UAS observed the component repository from a higher altitude to have a larger field of vision. If the component was identified successfully in this field, a rough position would be given. Next, depending on the rough position information, the UAS got close the component to acquire a more precise location. The whole positioning process is graphically described in Fig. 8.

Yaw adjustment of the UAS. The heading angle of UAS was required to be paralleled with the long side of the components to acquire the best grasping results. The actual attitude of the UAS was shown in Fig. 9(a), and the one in Fig. 9(b) was the required. One of the means to calculate the required angle of rotation was shown as followed:

$$\alpha = arcsin(\frac{2\sqrt{5}}{5} \cdot d) - arctan\frac{1}{2} \tag{4}$$

where:

$$d = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2}$$

or

$$d = \sqrt{(x_4 - x_2)^2 + (y_4 - y_2)^2 + (z_4 - z_2)^2}$$

(x_1, y_1, z_1), (x_2, y_2, z_2) , (x_3, y_3, z_3) , (x_4, y_4, z_4) were the coordinates of the four vertices of the enclosing rectangle acquired by the binocular camera. The process of yaw adjustment started as soon as the coarse positioning ended and followed by the fine positioning.

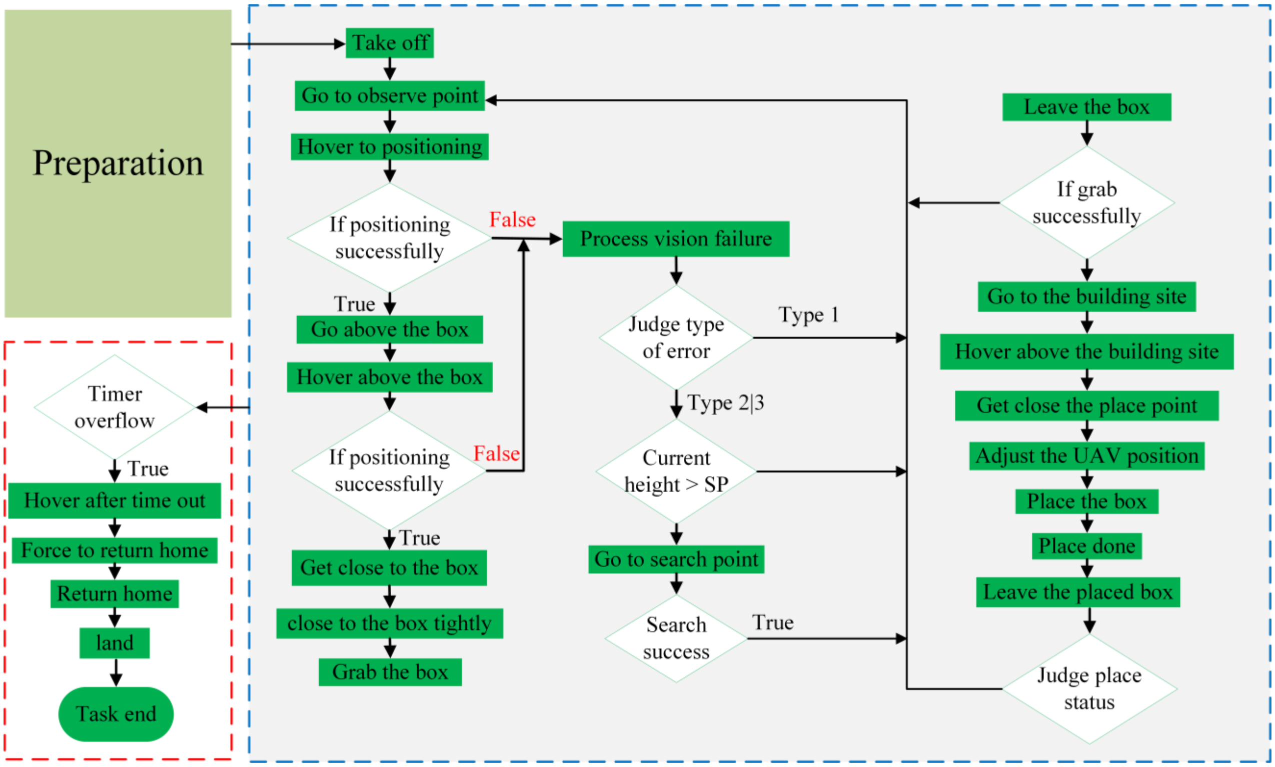


Fig. 10 An overview of the mission planner that describes the procedure of the mission.

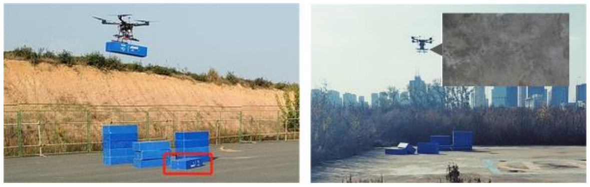


Fig. 11 Two common mistakes during object recognition and positioning.

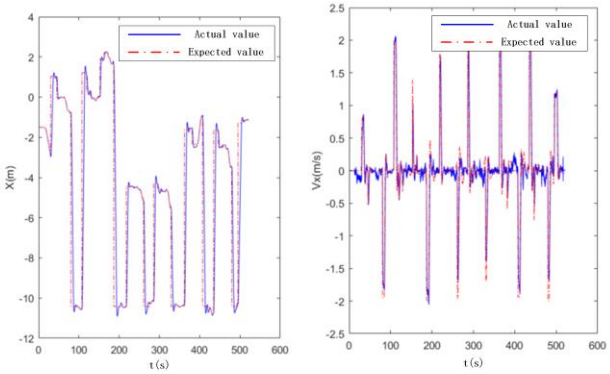


Fig. 12 The performance of the flight controller in actual flight.

D. Task Planning and Scheduling

It was the program that endowed the UAS with the autonomy to perform the mission, which was designed to supervise all of the other programs and modules to receive the return data of them and published commands to them. Fig. 10 gives an overview of the designed mission task procedure.

The program that played the role of task planning and scheduling was a rules-based state machine. Concretely speaking, it defined a series of states that the UAS might confront during performed the mission. For every possible state, corresponding rule in the if-then form was made to assign an appropriate action to the UAS. The rules were required to be complete, i.e. contained all possible rules.

The measurement of onboard sensors was used to determine the current state the UAS, such as the current position and attitude of the UAS and the coordinate information of the components. Twenty-nine states in total were defined based on the logical analysis and actual experiments, which could divided into the following important parts: 1) took off, 2) reached the components repository, 3) recognized and positioned, 5) grasped and judged, 6) transported, and 7) stacked, 8) repeated from 2) to 7), and 9) returned home once the time ran out.

E. Fault-tolerant Measures

The key requirement for software development was to ensure the robustness and reliability of the algorithms associated with the mission strategies in all operating situations, because no interaction were allowed throughout the entire competition. Some fault-tolerant measures were implemented to handle the logic error and hardware failure which would occur during the mission, which included the following parts.

The exception handling in object identification, which included object loss and target error, see Fig. 11. Object loss meant that there was no component within the field of the vision. The solution was to increase the altitude the UAS was

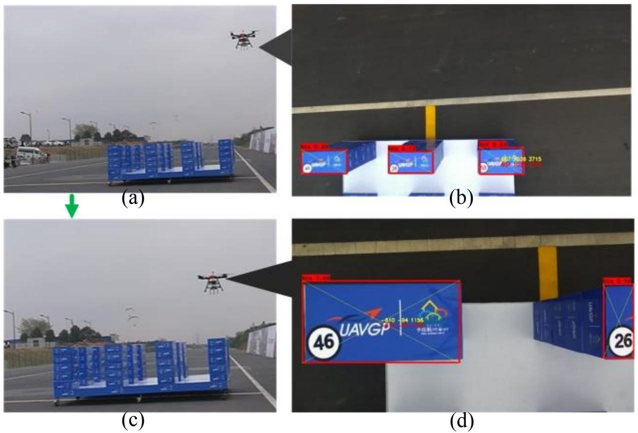


Fig. 13 The process of coarse positioning and fine positioning.

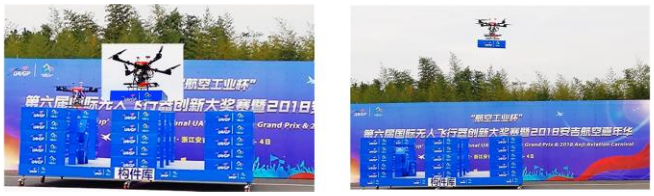


Fig. 14 The process that the UAV grasped the component autonomously.

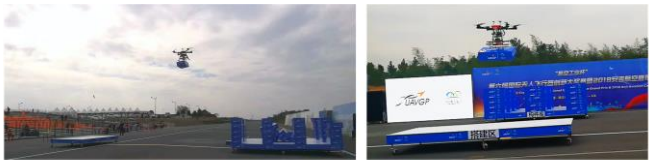


Fig. 15 The process of transporting and placing the component.

in to expend the field of vision, if this didn't work, then the box search mechanism would be triggered. The UAS would follow a predefined trajectory to search the box in the whole competition area until the one of them was obtained. While target error meant that the component was successfully recognized, but a very low altitude which might be dangerous for the UAS was provided. At this time, the UAS would ascend to the certain height to conduct the coarse positioning again until a new box was identified.

Grasping or placing components unsuccessfully was allowed. Inevitably, the situation that the UAS failed to grasp the components might occur, in which a distance sensor was used to avoid idle work. After the grabbing or placement action, the UAS would rise to a certain height at which to judge the if the component was grabbed or placed successfully or not, then the result was used to determine the UAS to grab or place again or take a new action.

The proception mechanism provided by ground station. The current situation was important for operators when the UAS was carrying out the mission tasks, based on which weather the UAS performed the mission as expected was determined. In addition, it could execute some emergent operations, such as status switching, to avoid some disastrous consequence. The ground station and the flight controller were connected by wireless data transmission module.

IV. Application of The Strategies

The strategies addressed in the previous section were implemented as real-time software on three different computers and applied to the UAS. And the performance of the software/hardware integrated UAS was tested through field experiments. Various settings and parameters of the UAS were tuned and updated to achieve a reliable and consistent performance. In the actual mission, the whole process was performed smoothly and intelligently.

A. Navigation and Control

The navigation based on the EKF fusion and the control based on cascade PID were implemented to perform the low-level autonomy of moving and hovering through the whole competition. Fig. 12 shows the position and velocity tracking with respect to x-axis in actual flight. The navigation and control algorithms showed stable and consistent performance throughout the competition, which was accurate enough for the UAS to perform the mission tasks with a positioning error less than ten centimeters.

B. Recognition and Location

To validate the design of the vision system which includes recognition and positioning, a number of field tests were

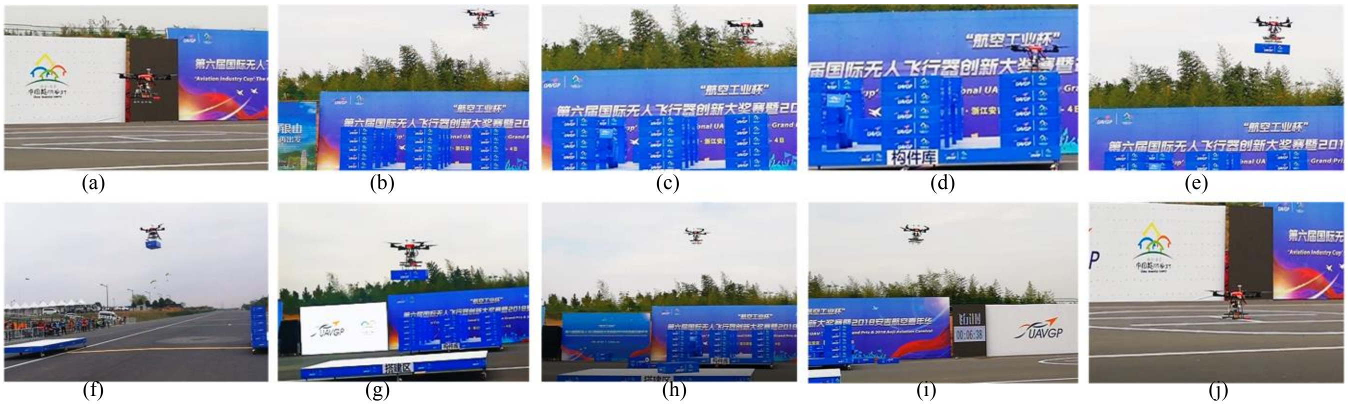


Fig. 16 The entire procure of the mission. (a) Take-off. (b) Coarse positioning and yaw adjustment. (c) Fine positioning. (d) Component capture. (e) Rise and grasp status judgement. (f) Transport. (g) Placement. (h) Return to grasp again. (i) Return home. (j) Landing.

performed. Fig. 13(a)(b) shows the coarse positioning from a high altitude and the image captured by the camera, while Fig. 13(c)(d) shows the fine positioning and the corresponding captured image. Here, the component with yellow coordinate information was the chosen one to grasp. The combination of the two achieved the intended effect, i.e. it not only expanded the recognition field of vision but also improved the positioning accuracy. Both in the test environment and competition environment, the vision system showed reliable and robust characteristic, which could provide accurate navigation information for the UAS to grab and place the component.

C. Grasp and Construction

Fig. 14 shows the process that the UAV grasped component autonomously. Generally, due to the high altitude the UAS was in, there were usually more than one box in the field of vision, under this situation, the component that had a shortest distance from the UAS would be chosen to grasp. Because the coordinate information transmitted to the flight controller was ten centimeters below the center of the component, which ensured an almost unbiased grasp. The telescopic sucker structure proved to be effective and efficient in practical application based on a high positioning and navigation accuracy. Here, the detection mechanism of whether the box was successfully fetched reduced wasted effort indeed.

Fig. 15 shows the process of transport and placement. Thanks to the high-precision control and navigation, the UAS had the capacity to locate at the predefined position with a high accuracy and the component could be placed successfully.

D. Task planning and scheduling

In the experiment, the task planning and scheduling was tested with respect to the logic and error handling. These required a number of field tests to find out potential logic error and some situations that hadn't been considered yet. After a series of logical analysis and algorithms updates, the task planning and scheduling could guide the UAS to complete the mission tasks autonomously with reliable, robust, and efficient performance. Fig. 16 demonstrates the main procedures of the entire mission.

V. Conclusion

This paper addresses the major components and the development process of the UAS from the design of hardware to the implementation of the software, which covered the entire spectrum of the mobile robotics. The hardware proved to meet the requirements of the mission, which ensured load capacity when had a enough endurance. And by designing a new grasp mechanism, the components which had large and flat surface could be grabbed effectively. The software algorithms provided accurate navigation and control, real-time perception and planning. And the fault-tolerant measures improved efficiency of grasp in the test and actual competition. All of the work endowed the UAS with the ability to perform the mission without manual intervention, which achieved a satisfied performance and outcome (the second award out of 13 teams). More broadly, the techniques used in the mission have potential real-world application, e.g. the UAS transport of the express industry. In the future work, some modification to improve the grasping accuracy should be considered, and some techniques to reject the wind disturbance will be introduced.

References

[1] G. Cai, J. Dias, and L. Seneviratne, “A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends,” *Unmanned Syst.*, vol. 02, no. 02, pp. 175–199, Apr. 2014.

[2] H. Shakhatreh *et al.*, “Unmanned Aerial Vehicles: A Survey on Civil Applications and Key Research Challenges,” *ArXiv Prepr. ArXiv180500881*, 2018.

[3] S. Nuske *et al.*, “Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers,” *J. Field Robot.*, vol. 32, no. 8, pp. 1141–1162, 2015.

[4] M. Rossi and D. Brunelli, “Autonomous gas detection and mapping with unmanned aerial vehicles,” *IEEE Trans. Instrum. Meas.*, vol. 65, no. 4, pp. 765–775, 2016.

[5] S. Gupte, Paul Infant Teenu Mohandas, and J. M. Conrad, “A survey of quadrotor Unmanned Aerial Vehicles,” in *2012 Proceedings of IEEE Southeastcon*, Orlando, FL, USA, 2012, pp. 1–6.

[6] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[7] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, “A complementary filter for attitude estimation of a fixed-wing UAV,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 340–345.

[8] S. Madgwick, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” *Rep. X-Io Univ. Bristol UK*, vol. 25, pp. 113–118, 2010.

[9] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, 2011, pp. 1–7.

[10] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, vol. 1, pp. I–I.

[11] R. Brunelli and T. Poggio, “Face recognition: Features versus templates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 10, pp. 1042–1052, 1993.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886–893.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[14] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I–I.

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

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[17] W. Liu *et al.*, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, 2016, pp. 21–37.

[18] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *ArXiv Prepr. ArXiv180402767*, 2018.