Design of Autonomous Aerial Robot System

Min-Fan Ricky Lee, Cheng-Chia Lee, Meng-Ying Chuang
Graduate Institute of Automation and Control,
National Taiwan University of science and Technology, Taipei, Taiwan
Tai-Lin Chin, Yu-Chuan Huang, Chieh-Sheng Lin, Ya-De Chen
Department of Computer Science and Information Engineering,
National Taiwan University of science and Technology, Taipei, Taiwan
Xiao-Fei Lu
Department of Mechanical Engineering,
National Taiwan University of science and Technology, Taipei, Taiwan
Pei-Cheng Chung
Department of Industrial and Commercial Design,
National Taiwan University of science and Technology, Taipei, Taiwan

ABSTRACT

A hierarchical and distributed UAV system is proposed including hardware, software and appearance design. For the hardware part, a quadcopter is adopted as the UAV locomotion platform. APM (ArduPilot Mega) is used for the low level flight pose control. An arduino embedded system plays the role of the interface between the onboard low level controller and the ground high level controller. Onboard sensors include compass, magnetometer, IMU (gyroscope and accelerometer), barometer, sonar and two CCD cameras. For the software part, a fuzzy logic control system is proposed for the high level autonomous navigation. The optical flow, Lucas-Kanade approach, is exploited for the localization of the UAV and the ground mobile robot through the images provided by the bottom camera. The optical flow, Gunnar Farneback approach, is applied for obstacle avoidance with the frontal camera. In particular, an innovative UAV appearance is designed and built. The aerodynamic and UAV propeller protection are the primary design considerations while not affecting the onboard sensing and motion.

Index Terms—Unmanned Aerial Vehicle; aerial robot; visual navigation; appearance design; autonomous; obstacle avoidance

1. INTRODUCTION

This paper propose an autonomous aerial robot system aimed for the 2015 The International Aerial Robotics Competition (IARC) - 2015 Symposium on Indoor Flight Issues. The symposium topics as follows

- Systems for Flight in Confined Areas
- Autonomous Flight Control in Micro Air Vehicles
- Close Quarters Obstacle Detection and Avoidance
- Indoor Navigation Apart from GPS
- Low Power Communication in Spectrally-Cluttered Environments
- Survivable Airframes and Propulsions in Obstacle-Rich Environments

  The proposed system in this paper is to address the challenge in Mission 7 of IARC are to demonstrate three behaviors
- "Interaction between aerial robots and moving objects (specifically, autonomous ground robots).

- Navigation in a sterile environment with no external navigation aids such as GPS or large stationary points of reference such as walls.
- Interaction between competing autonomous air vehicles.

The wireless networked control, highly distinctive feature descriptors (HDF) and a visual odometry system are proposed in a vision-based navigation system for small unmanned aerial vehicles (UAVs) is proposed [1]. The visual odometry system uses a single camera to reconstruct the camera path and the structure of the environment.

A generalized predictive control in a networked control system was proposed to compensate for the network delay and packet loss. It demonstrates the feasibility of real-time closed-loop control in aerial robot navigation [2].

A real-time and integrated multipose face tracking and recognition system mounted on an unmanned aerial vehicle (UAV) in a search and rescue operation is proposed [3].

An aerial robots cooperation system, high and low altitude is proposed for autonomous navigation and landing [4]. The Image Based Visual Servo (IBVS) is applied for tracking the landing site and landing motion. The Intelligent Control System is designed for the high level posture control. The results demonstrate shows an autonomous and robust landing can be achieved under multiple robot collaboration.

A visual servo navigation is proposed for UAV (unmanned aerial vehicle) and GMR (ground mobile robot) collaboration. The UAV hover at an optimal attitude for visual perception, ground mapping and localization of GMR, target and obstacles. The shortest collision free path is generated for the GMR to arrive the target. A hierarchical networked control system is also proposed [5].

An aerial and ground robotic collaboration system was proposed for the search and rescue mission in disaster. The joint operation achieved the detection and avoidance of obstacles with different features to guide ground robot to target by aerial robot [6].

1.1. Problem statement

In order to achieve the mission of International Aerial Robotics Competition which includes several task, this paper designs a system for autonomous unmanned aerial vehicle (UAV). The goal is to build a UAV system that can demonstrate the ability of stable flight and interaction with ground robot without external navigation sensors such as GPS.

1.2. System Overview

The proposed system is illustrated in Figure 1. The autonomous aerial robot is used to drive the ten target robots pass the goal line. When the aerial robot decides to change the moving direction of the ground robot, it descends and trigger the Hall Effect sensor on the top of the ground robot. Four other ground robots with tall cylinders are used as moving vertical obstacles and will change the motion of the other ten target robots randomly when the target robots hit the four cylindrical robots.

1.3. Conceptual Solution

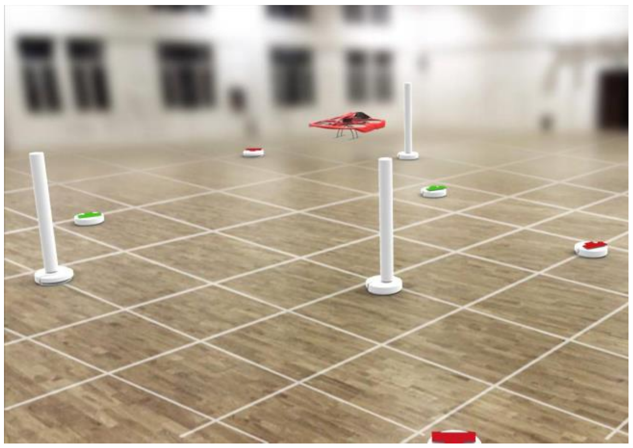


Figure 1. Schematic diagram

A UAV is designed and built from scratch. The major design includes hardware, software and the appearance. The UAV hardware components are chosen according to the mission requirements. Sensors such as cameras, inertia measurement unit (IMU) and sonar are used to sample the environment and provide information for reacting to the physical situation instantly. For the software design, a number of algorithms are developed for the specified mission such as visual navigation, optical flow based tracking, and localization with no GPS. In particular, the appearance design provides an innovation design concept for UAV, which considers aerodynamics, protection mechanisms for the propellers, and cost. From simulations, the total solution of the UAV performs well and is expected to complete the specified mission.

The rest of the paper is organized as follows. The concept of the whole system design is introduced in Section 2. Section 3 presents emulation results. Section 4 concludes the paper.

2. Method

2.1. Hardware Design

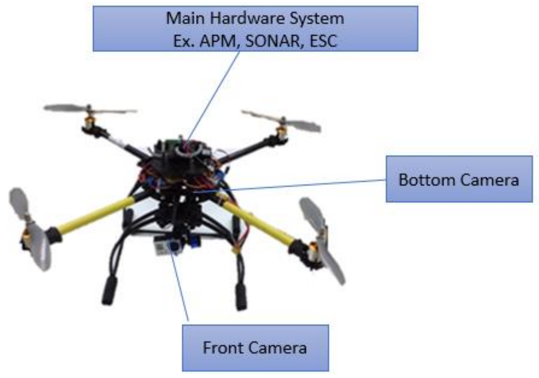


Figure 2. Aerial robot

Figure 2. shows the final assembled UAV and Figure 3. is the designed harward architecture. APM is chosen as the flight control board.It includes a MCU, an IMU, and a barometer in a multi-function controller board. The electronic speed controller controls the motor speed by the pulse width modulation signal from the APM. The IMU is a three-axis attitude angle or anguler rate measuring device. It will be equipped with a three-axis gyroscopes and accelerometers to measure the angular velocity and acceleration of objects in three-dimensional space, which are used to calculate the object attitude. Generally IMU is mounted on the center of the UAV. The camera captures images transmitted to the ground station via the image transmitter. The sonar includes an ultrasonic transmitter, a receiver and the control circuit. When it is triggered, it will launch a series of 40 kHz ultrasound and receive echo from the nearest object. The

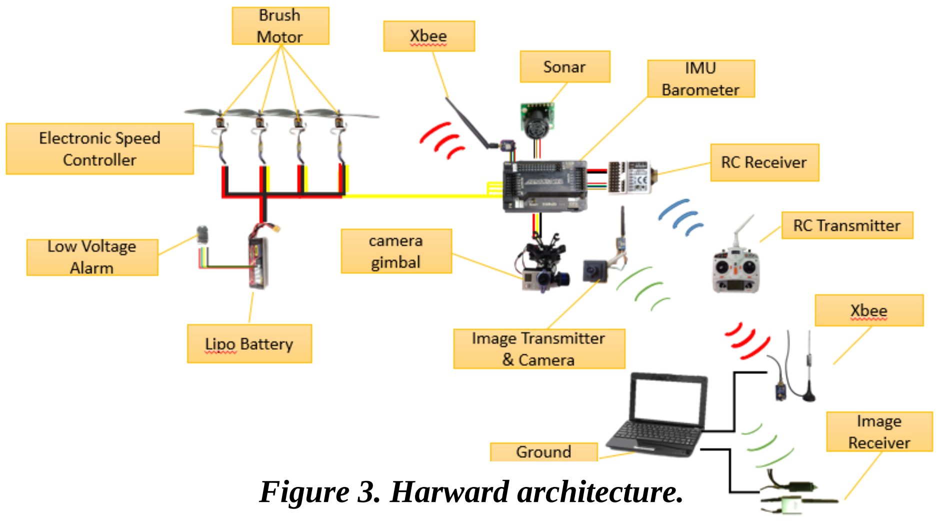


Figure 3. Harward architecture.

distance between the sonar and the object is measured by the elapsed time of the echo. Speed of sound in air is about 340 meters per second. Therefore, the sound propagation time for 1 cm is 1,000,000 / (340 * 100) = 29.4 microseconds. Note that the elapsed time is the round trip time and should be divided by two for the distance calculation. The sonar is used to calculate the height of the UAV. Finally, Xbee is used to connect the UAV and the ground station.

2.2. Algorithm

2.2.1. Gradient based Optical Flow Estimation

The optical flow is adopted in the proposed system for tracking the motion of the target robots. According to the homogeneous assumption of the image, the intensity of the tracking target keeps constant along the motion trajectory, satisfying $\frac{dI(x,y,t)}{dt} = 0$ as

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t) \tag{1}$$

Where

$I(x,y,t)$	Intensity of pixel *P*(*x*, *y*) at time *t*
$\Delta x, \Delta y, \Delta t$	Displacement of *P*(*x*, *y*) in *x*, *y*-axis with time elapse Δ *t* respectively

Through Taylor expansion, Equation (1) becomes

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + + \frac{\partial I}{\partial t}\Delta t + \cdots \tag{2}$$

The higher order term is neglected, and set

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0. \tag{3}$$

As $\Delta t \rightarrow 0$,

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}\Delta t = 0 \tag{4}$$

$$I_x u + I_y v + I_t = 0 \tag{5}$$

Where

u	$\frac{dx}{dt}$, *x*-axis component of optical flow vector
v	$\frac{dy}{dt}$, *y*-axis component of optical flow vector
I_x, I_y, I_t	The partial derivative of intensity of the reference pixel along the *x*, *y*, *t* direction

Equation (5) becomes

$$\nabla I \cdot U + I_t = 0 \tag{6}$$

Where

∇I	(I_x, I_y), Gradient direction
U	$(u,v)^T$, Optical flow

Lucas-Kanade approach assumes the motion optical vector is constant in a small neighborhood *Ω* of (x, y) as the constraint. The weighted least square is applied for the estimation of the optical flow. The error is defined as

$$\sum_{(x,y)\in\Omega} W^2(x,y)\left(I_x u + I_y v + I_t\right)^2, \tag{7}$$

where $W^2(x,y)$ is the weight window function.

This weight increases the effect of the local area to the constraint. Thus, Equation (7) becomes

$$U = (A^T W^2 A)^{-1} A^T W^2 B \tag{8}$$

Where

A	$[\nabla I(X_1), \cdots, \nabla I(X_n)]^T$

$W \qquad diag[W(X_1), \cdots, W(X_n)]^T$
$B \qquad -[\nabla I(X_1), \cdots, \nabla I(X_n)]^T$

Gunnar Farneback approach is using the idea of polynomial expansion to approximate some neighborhood of each pixel with a polynomial. The method can only be used on quadratic polynomials, which gives the local signal model, expressed in a local coordinate system.

$$f(x) \sim xTAx + bTx + c \qquad (9)$$

Where
$\boldsymbol{A}$ is a symmetric matrix,.
$\boldsymbol{b}$ is a vector and .
c is a scalar.

The coefficients are estimated from a weighted least square fit to the signal values in the neighborhood. The weight has two components called certainty and applicability. These terms are the same as in normalized convolution, which polynomial expansion is based on. The certainty is coupled to the signal values in the neighborhood. The applicability determines the relative weight of points based on their positions in the neighborhood. Typically one wants to give highest weight to the center point and let the weights decrease radially. The width of the applicability determines the scale of the structures which will be captured by the expansion coefficients [1].

2.2.2. Obstacle Avoidance

First, optical flow vectors are computed from image sequence. To make a decision about the robot orientation, the calculation of the focus of expansion (FOE) position in the image plane is necessary, because the control law is given with respect to the FOE. Then, the depth map is computed in order to know whether the obstacle is close enough to need immediate attention, or is in a distance to give an early warning or is far away from the robot that the robot can ignore it. The detail steps are described as follows:

- **Feature Selection**

In this algorithm, corners which can be identified in two consecutive frames are found in a frame. We use one common feature selection algorithm call Harris Corner detection in OpenCV.

- **Tracking and Correspondence**

In order to compute the optical flow vector of a point between two frames, the location of this point in the corresponding frames must be determined. The point has to be determined in all frames. The optical flow tracking algorithm based on Gunnar Farneback approach in OpenCV is used.

- **The Balance Strategy**

The balance strategy is based on a concept called motion parallax. Motion parallax compares the angular position change in two consecutive frames of a particular line which connects two fixed points. The angular position change is due to the motion of the observer. As the robot move, distant objects appear to move more slowly than objects that are close by. Consequently, motion parallax can provide the depth of an object. For a moving robot, closer objects will produce a larger optical flow than those that are further away. The region is divided into the left and right sides. To avoid obstacles, the robot moves away from the side with larger flow to the side with lower flow [2].

2.2.3. Visual Odometry based on Optical Flow

The onboard bottom camera is used to get frames. Harris Corner detection is implemented to get distinct points. In this phase of algorithm, Pyramidal Lucas-Kanade approach is used to track the points. The movement of specific points between two frames is calculated in pixel. Altitude information is obtained from sonar sensor which is on the bottom as well. Afterward, the movement which is in the image could be calculated by field of view (FOV). Then comparing to the altitude, the real displacement is obtained by trigonometric functions.

2.2.4. IMU Odometry

IMU is used to localize the robot. Robot needs to find obstacles, avoid them, and then plan the optimal path. One common method is to use inertial measurement units (IMU), which combines a 3-axis accelerometer and a 3-axis gyroscope. To be specific, an accelerometer is a device that monitors the acceleration of robots while a gyro provides information of rotation and orientation.

The measured speed of a mobile robot is valid if it is on a horizontal plane. However, when the robot moves on an inclined plane, deviations of the robot speed cannot be avoided because of the weight component caused by gravity. In that case, the error should be corrected.
Two reference frames are built to solve this problem. One XYZ coordinate system is used for the inertial frame and the other xyz coordinate system is used for the frame, which is constructed for the center of mass of a robot. Then, we use measured data (the acceleration of the xyz frame $\ddot{\vec{r}}_{xyz}$; standard Euler angle θ_x, θ_y and θ_z; skew symmetric angular velocity ω_x, ω_y, ω_z and angular acceleration matrices $\dot{\omega}_x$, $\dot{\omega}_y$, $\dot{\omega}_z$) to complete the following calculation.

$$\mathrm{J} = \begin{pmatrix} \cos\theta_y \cos\theta_z & -\sin\theta_z \cos\theta_x + \sin\theta_x \sin\theta_y \cos\theta_z & \sin\theta_x \sin\theta_z + \cos\theta_x \sin\theta_y \cos\theta_z \\ \cos\theta_y \sin\theta_z & \cos\theta_z \cos\theta_x + \sin\sin\theta_y \sin\theta_z & -\sin\theta_x \cos\theta_z + \cos\theta_x \sin\theta_y \sin\theta_z \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_x \cos\theta_y \end{pmatrix} \quad (10)$$

$$\omega = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (11), \qquad \dot{\omega} = \begin{pmatrix} 0 & -\dot{\omega}_z & \dot{\omega}_y \\ \dot{\omega}_z & 0 & -\dot{\omega}_x \\ -\dot{\omega}_y & \dot{\omega}_x & 0 \end{pmatrix} \quad (12)$$

$$\ddot{\vec{r}}_{xyz} = \mathrm{J}\,\ddot{\vec{r}}_{XYZ} + 2\dot{J}\,\dot{\vec{r}}_{XYZ} + \ddot{J}\,\vec{r}_{XYZ} \quad (13)$$

Therefore, the actual vehicle acceleration components $\ddot{X}$, $\ddot{Y}$, and $\ddot{Z}$ related to the accelerometer measurements $\ddot{\vec{r}}_{XYZ}$ through

$$\begin{pmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{pmatrix} = J^T \left(\ddot{\vec{r}}_{XYZ} - 2\dot{\omega} J\,\dot{\vec{r}}_{XYZ} - (\dot{\omega} + \omega^2) J\,\vec{r}_{XYZ} \right) + \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} . \quad (14)$$

Then the actual speed and displacements can also be obtained through integration [3].

2.2.5. Target Robot Path Planning

Visibility graphs are applied to find Euclidean shortest paths among a set of polygonal obstacles in the plane [5]. Let *S* be a set of simple polygonal obstacles in the plane, then the nodes of the visibility graph of *S* are just the vertices of *S*, and there is an edge (called a visibility edge) between vertex *v* and *w* if these vertices are mutually visible.

$$S^* = S \cup \{P_{start}, P_{goal}\} \quad (15)$$

Where

P_{start} is the start point.

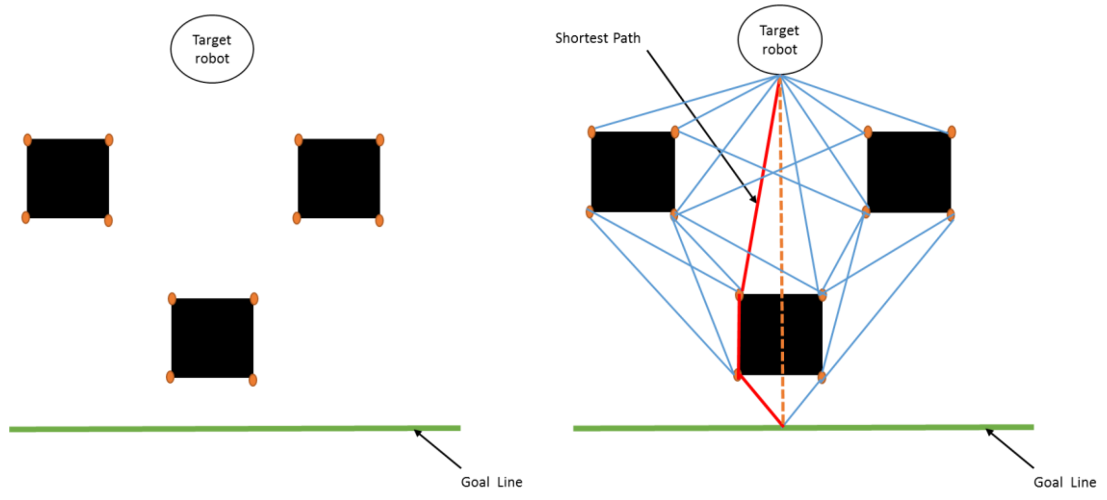
P_{goal} is th goal point.

Algorithm VISIBILITYGRAPH(S)

```
    Input. A set $S$ of disjoint polygonal obstacles.
    Output. The visibility graph $G_{vis}(S)$.
1. Initialize a graph $G = (V, E)$ where $V$ is the set of all vertices of the polygons in $S$ and $E = \varnothing$
2.  for all vertices $v \in V$
3.  do $W \leftarrow$ VISIBLEVERTICES($v$, $S$)
4.  For every vertex $w \in W$, add the arc $(v, w)$ to $E$
5.  return $G$
```

Localization, path planning and obstacle avoidance are the primary tasks in IARC competition. In this section, the visibility graph and Dijkstra algorithm which are the basis of the proposed method, and briefly explained as Figure 4. The Dijkstra algorithm solves the shortest-paths problem with weighted graph G = (V, E) for the case in which all edge weights are non-negative, which is used to find the path with lowest cost (i.e. the shortest path) between the source vertex and destination on the graph.

Obviously, the shortest path in Figure 4. (b) is the dash line which is connecting the start point and the goal point (called S-G line) if and only if it does not penetrate any obstacles. Otherwise, the shortest path is found in the visibility graph by Dijkstra algorithm as in Figure [6].



(a) ***(b)***

Figure 4. (a)Detecting all the vertexes of obstacles (b) Connecting all the vertexes and finding a shortest path by Dijkstra algorithm

2.2.6. Strategy of scenario

The basic flying control procedure is in Algorithm Navigation. The basic strategy is that the UAV will drive the ground robots to the aim zone one by one. This simple strategy is easy to implement without complex computation, which is a necessary and desired design characteristic by our UAV micro controller. Specifically, in each driving iteration, the UAV will search a target robot and fly to the top of the robot using our obstacle avoidance aviation control method. Then, the UAV will start to monitor the moving direction of the target robot. If the angle θ of the line from the robot to the aim zone and the moving direction, as shown in Figure 5, is greater than 45 degrees, the UAV will descend and touch the target robot until the robot turns to the direction toward the aim zone. The UAV will keep monitoring the target robot until it goes into the aim zone. Then, the UAV will search another target robot and repeat the procedure.

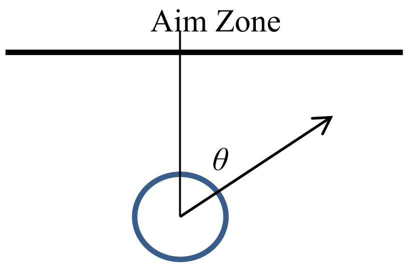


Figure 5. Moving direction monitoring for the targeted ground robot

Two strategies for searching the target robot are proposed. The first is to find the one which is closest to the aim zone. The potential superiority is that the monitoring mechanism is simple since only one robot is monitored and may score sooner. The other one is to find the one with the furthest distance from the aim zone and drive it until it reaches the aim zone. In this strategy, there may be ground robots entering the aim zone by themselves without any guiding efforts. However, it may consume more time to score.

Algorithm Navigation

```
1.  Search the target ground robot based on the selected search target strategy
2.  Fly to the top the targeted ground robot by obstacle avoidance aviation
3.   IF the targeted ground robot is inside the arena
4.        Monitor the moving direction of the targeted ground robot
5.        IF the angle θ >  45 degrees
6.              Fly downward and touch the targeted ground robot
7.        End IF
8.        Fly upward and goto step 3
9.  ELSE
10.       Goto step 1
11. End IF
```

2.3. Software Design

Figure 6. shows the major structure of the system. Basically, the system is divided into four modules. The function of each module is briefly described as follows:

- Controller: The controller is brain of the system. It uses the data from the sensors, like the height estimate from the sonars, and the results from the image processing module, like the selected targeted ground robot location, to direct the navigation of the UAV. The major jobs of the Controller is to do the gradient based optical flow estimation, targeted robot path planning, and control the UAV aviation.
- Image processing module: This module provides the functions relative to images. It reads data from camera and provides services to the Controller. The services include finding the targeted ground robot location, monitoring the robot moving direction, providing the current UAV location, and identifying obstacles.
- UAV control module: This module provides the interface to control the UAV hardware. It provides the basic commands to the UAV hardware, for instance, rising up, flying forward, turning right, turning left, and flying downward.

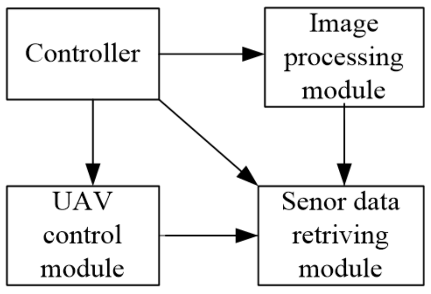


Figure 6. The system structure

- Sensor data retrieving module: This module reads data from sensors including the camera, the sonar, and the IMU.

The control activity diagram is shown in Figure 7. The flow chart shows the major activities running in the Controller module. Essentially, the Controller will initialize all sensors and start rising up. The sensor data retrieval process will start and run simultaneously with the Controller, so that the other modules and the Controller can read instant data from the sensors. The Controller, then, finds the targeted ground robot location using the services provided by the image processing module and flies to the top of the robot. It continues to monitor the moving direction of the targeted ground robot and drive it into the aim zone. Then, the Controller will find the next targeted ground robot and repeat the same procedure until no ground robots left in the arena.

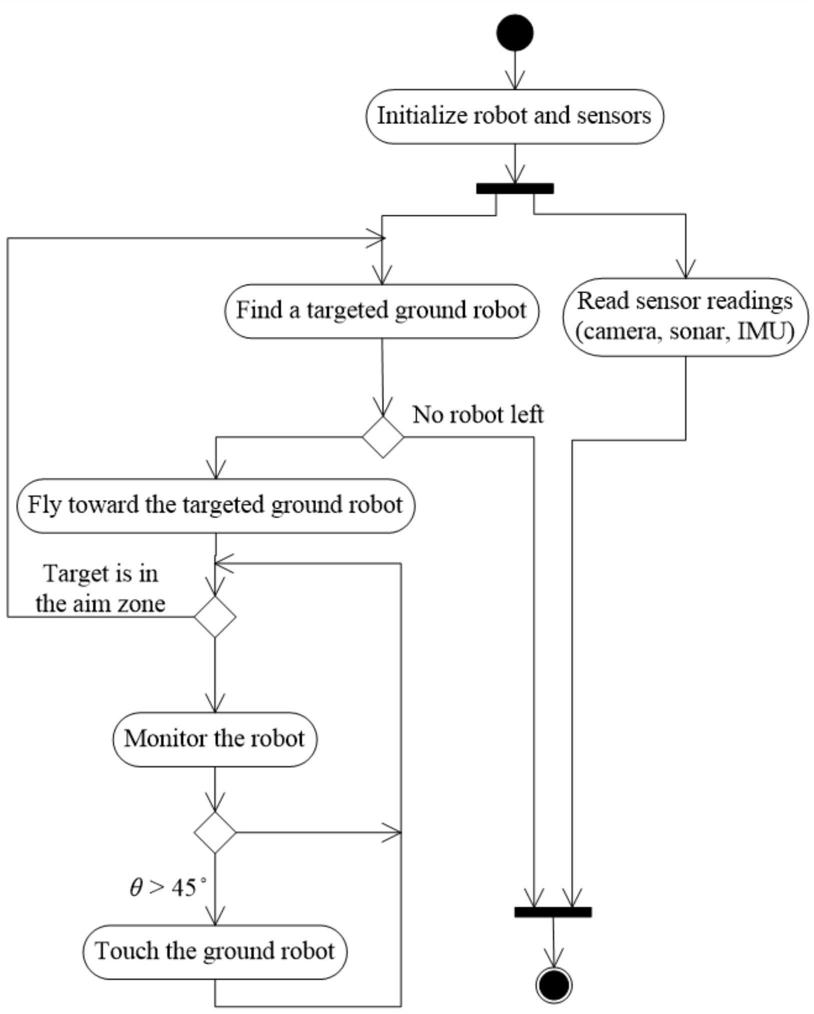
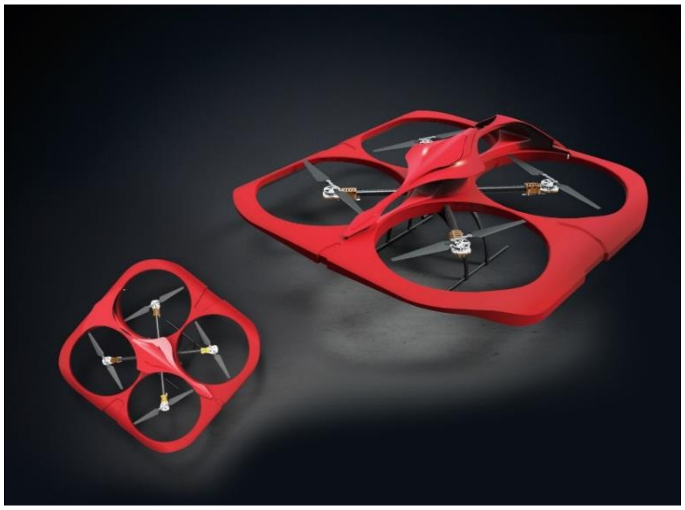

Initialize robot and sensors
Find a targeted ground robot
Read sensor readings (camera, sonar, IMU)
No robot left
Fly toward the targeted ground robot
Target is in the aim zone
Monitor the robot
$\theta > 45^\circ$
Touch the ground robot

Figure 7. The activity diagram of the control system

2.4. Appearance Design



- **Design Concept**

This design is inspired by dragon's scales, which means protection, fast flying, and oriental elegance and nobility. The body is made of PS and the top shell is "scales", slightly overlapping each other and creating a sophisticated decorative effect. To fulfill the goal and mission, Dragon scales are very reliable and just like a wing with a solid shell.

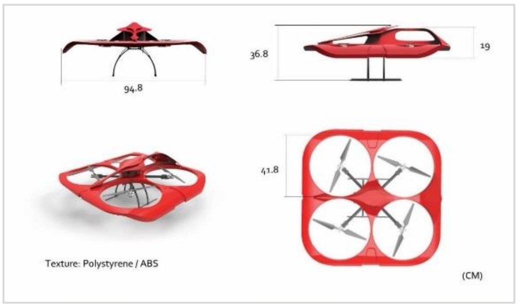
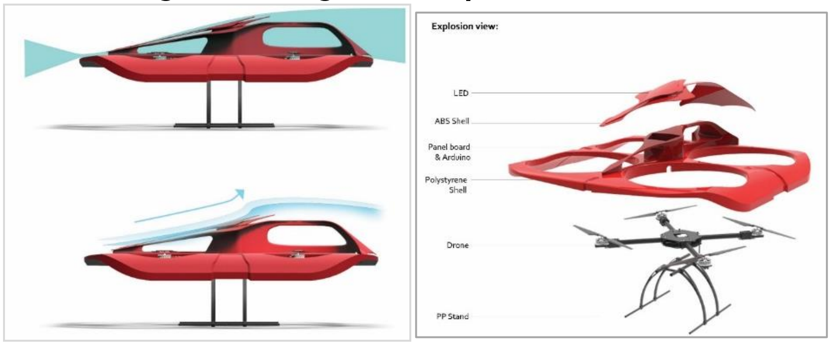


Figure 8. Three dimensions and size

Figure 9. Design Detail of the drone shell



(a) *(b)*

Figure 10. (a)Reducing wind resistance (b) Explosion view

3. Result

3.1. Obstacle Avoidance

The image is divided into the left and right side as shown in Figure 11. The sum of the magnitude of optical flow on each side is first computed. The robot then makes a decision for the next direction to avoid the obstacle. The decision is based on sum of the optical flow values in each side. The sum of magnitude in the left side is smaller than that in the right side. Thus, the robot will turn to the left.

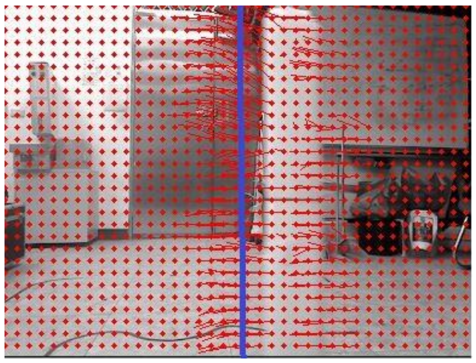
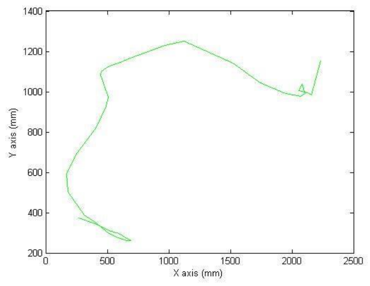
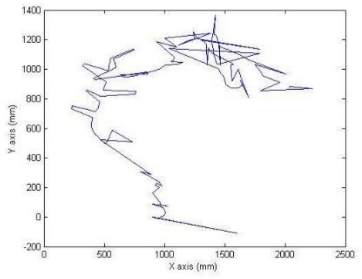


Figure 11. Image of a frame showing Optical Flow magnitude.

3.2. IMU Odometry and Visual Odmertry base on Optical Flow

The results are shown in Figure 12. (a) and (b). As you can see the path recorded by IMU is smoother. In contrast, the path recorded by optical flow method using camera is quite zigzag. However, the two methods could be combined. Afterward, the precise data can be obtained. In addition, optical flow by Lucas-Kanade approach is not only used for visual odometry, but also used to track the ground mobile robot.





(a) ***(b)***

Figure 12. (a)Trajectory recorded by IMU Odmertry. (b) Trajectory recorded by Visual Odmertry base on Optical Flow.

3.3. Real-Time Path Planning with Dynamic Obstacle

In this experiment, an experimental environment is constructed similar to the IARC competition scenario which includes target robots, obstacle robots and goal line as in Figure 13 (a). The purpose of this experiment is to verify that the algorithm of detecting and tracking the mobile obstacles is correct, which means the UAV can guide the target robot safely pass the goal line. The scenario is using UAV to track target robot until an obstacle appears in the searching zone. Two situations may occur. The first situation is that the obstacle is another ground robot. The robot speed can be obtained by the obstacle avoidance algorithm based on optical flow. If the ground robot could affect the motion of the target robot, the UAV will land in front of the ground robot and let the original target robot stay on its path. Otherwise, the UAV will ignore the ground robot. The other situation is the obstacle is a cylindrical robot. If the cylindrical robot is far enough, the UAV will touch the target robot to avoid the cylindrical robot. Otherwise, a new path will be generated after the cylindrical robot hits the target robot.

3.4. Visual system detect target and obstacle robot

The ground robots and cylindrical robots are determined by the following two steps. The first step is using the magnitude and direction of the optical flows in the field to determine moving objects. Then, the ground robots and the cylindrical robots are differentiated by color. The following equations capture the locations of obstacles, free space and goal point based on HSV color module.

$$Obstacle(x,y) = \begin{cases} 255, & if\ src(x,y) > robots\ threshold \\ 0, & others \end{cases} \tag{16}$$

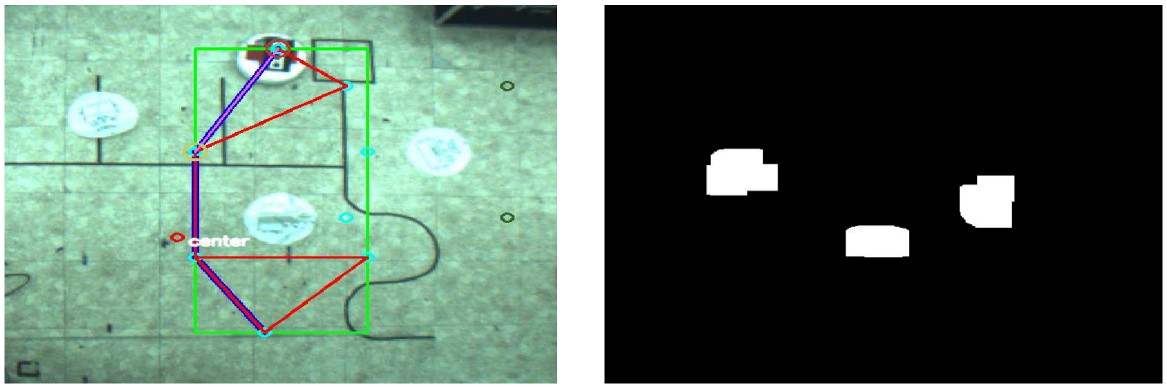
$$Free\ Space(x,y) = \begin{cases} 0, & if\ Obstacle(x,y) = 255 \\ 255, & others \end{cases} \tag{17}$$

$$Goal(x,y) = \begin{cases} set\ goal\ point, & if\ src(x,y) > green\ line\ threshold \\ 0, & others \end{cases} \tag{18}$$

Although the above method can detect ground robot, the image could contain some noises. Therefore, morphological operations are used to eliminate the noise by equation 19 and 20. The erosion operation on the input binary image can reduce the effect of noise. The dilation operation could extend obstacle shape to ensure that the UAV would not run into it. Figure 13.(b) displays the resulting image from Figure 13.(a) through image binarization and morphological processing.

$$E(A,B) = A \ominus B = \cap_{b \in B} (A - B) \tag{19}$$

$$D(A,B) = A \oplus B = \cup_{b \in B} (A + B) \tag{20}$$



(a) (b)

Figure 13. (a) Path planning with dynamic obstacles (b) Obstacle robot detection by morphological image processing

4. Conclusion

In this paper, UAV design considerations include effective and low-cost hardware, efficient software and innovative appearance as well as protection mechanisms. For UAV motion control, Robot navigation based on visual servo system and image processing for obstacle avoidance are exploited. From the simulations, our algorithms perform effectively for the specified mission. It is highly expected that the system will complete the mission in the competition.

Acknowledgment

The authors gratefully acknowledge use of the equipment and facilities of the Autonomous Robotic Laboratory at the Graduate Institute of Automation and Control in National Taiwan University of Science and Technology (NTUST). This competition attendance was financially supported by the NTUST "The Aim for the Top University Project" grant.

References

[1] Farnebäck, Gunnar. "Two-frame motion estimation based on polynomial expansion." Image Analysis. Springer Berlin Heidelberg, 2003. 363-370.

[2] Souhila, Kahlouche, and Achour Karim. "Optical flow based robot obstacle avoidance." International Journal of Advanced Robotic Systems 4.1 (2007): 13-16.

[3] Nistler, Jonathan R., and Majura F. Selekwa. "Gravity compensation in accelerometer measurements for robot navigation on inclined surfaces." Procedia Computer Science 6 (2011): 413-418.

[4] Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars "Computational Geometry Algorithms and Applications", 3rd rev. ed. Springer-Verlag Berlin Heidelberg, pp. 323-333, 2008.

[5] Min-Fan Ricky Lee and Chia-Hsien Louis Chen, "Dynamic Global Path Planning Based on Kalman/Particle Filter"