Autonomous Mini-UAV for Indoor Flight with Embedded On-board Vision Processing as Navigation System

Swee King Phang *, Jun Jie Ong *, Ronald T. C. Yeo *, Ben M. Chen * and Tong H. Lee *
* Department of Electrical and Computer Engineering, National University of Singapore, Singapore,
Email: {u0606584,u0606702,u0705959,bmchen,eleleeth}@nus.edu.sg

***Abstract*—This paper describes the development of an Unmanned Aerial Vehicle (UAV) with the aid of a vision processing system for indoor navigation. A co-axial radio-controlled (RC) helicopter is upgraded with a customized on-board avionics system which include two Gumstix Linux computer systems as the on-board processors. A camera module is used to capture real time video during the flight to perform real time image processing. High level control commands are sent to the avionics system such that the UAV could perform simple indoor navigation by tracking the colored tracks on the ground. Important data such as attitude, velocity and acceleration of the UAV, together with the real time video will be feedback to the ground station via communication links for giving commands and monitoring purposes. Flight tests have been carried out to verify the results of the vision processing system and to ensure the robustness of the controller in the UAV.**

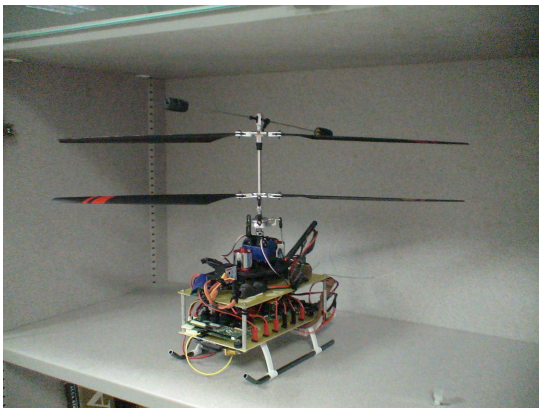

Fig. 1. KingLion in display.

I. Introduction

Driven by the rapid development of microprocessors, sensors and actuators, unmanned aerial vehicle (UAV) are getting smaller and lighter but with more sophisticated functions. It's tiny size realized the ability to navigate in the indoor environment. It can be widely used in remote observation of hazardous environment which is not accessible by other unmanned vehicles [1]. Besides obvious military applications, mini UAVs also serve as an excellent platform for researchers to investigate robust control theories [2] [3] [4].

The conventional control methodology involving the usage of GPS signal can no longer be realized in an indoor environment. In this paper, a vision based navigation system is proposed to replace the conventional GPS navigation system. It involves the usage of vision sensors such as camera and the implementation of on-board vision processing algorithms.

Vision processing is typically a costly operation that requires large amounts of computing power. This computational requirement usually translates to bulky processing units that can be carried only by large UAVs [5]. A common attempt to vision-enable mini UAVs is to include a control station whereby the video feed is sent to a fixed control station and processed results are sent back to the robots [6] [7]. However, our first approach using a ground control station is tested with limited success due to the interferences and delays in communication between control station and the UAV.

To overcome this problem, we proposed the used of on-board vision processing technology with simplified vision algorithms. A popular embedded vision system, CMUCam3, is capable to run simplified algorithms with reduced throughput and accuracy. However, we refused to sacrifice on processing throughput and hence found existing embedded vision systems lacking. The solution was to put together an embedded vision system that is more powerful than the current state of art and is still small and lightweight enough for our UAV.

A typical UAV should consist of the following essential parts [8]:

1) Physical aircraft with engines
2) An on-board avionics system which include
   a) On-board processor
   b) Inertia measurement unit (IMU)
   c) Communication system
   d) Power supply system
3) A ground control station.

Our UAV rotorcraft platform, KingLion is designed using these guidelines. The avionics system includes two processors, an inertia measurement unit, an ultrasonic sonar range finder, modems for communication purposes and a power supply system which last up to ten minutes flying time. There will be both data link and video link between the on-board processors and the ground supporting station.

To test the feasibility of the vision based navigation, the UAV is required to follow a colored track which has been painted on the ground as shown in Figure 2

This paper will be organized as follows: The overall

978-1-4244-7626-8/10/$26.00 ©2010 IEEE

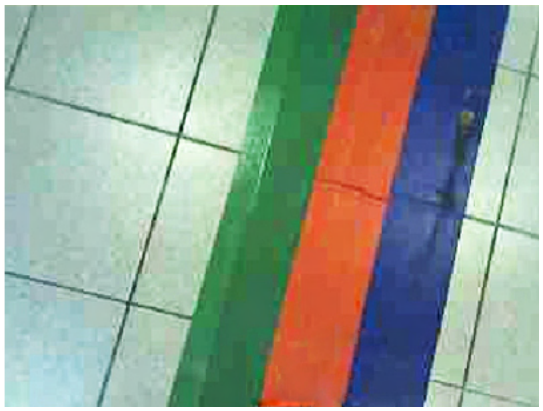

Fig. 2. Colored tracks on the ground. The ordering of colors determine the direction of travel

Fig. 3. ESky Big Lama Co-Axial Helicopter

platform design of KingLion will be shown in section II. section III and section IV presents the control and vision processing platforms respectively. section V covers the overall control scheme including vision processing and control algorithms. Various actual experimental results will be shown in section VI and finally our conclusions together with potential future improvements will be discussed in section VII.

II. Platform Design

A. Basic Helicopter

ESky Big Lama Co-Axial helicopter (Figure 3) is chosen to be the basic helicopter. It is a low cost toy co-axial helicopter with rotor diameter of 460 millimeters and weight of 410 grams. However, the effective payload of the original helicopter is not sufficient to carry the on-board avionics system. To increase the takeoff weight, the motors of the helicopter are upgraded to 3000KV brushless motors to increase torque and power. Stiffer blades are used to create higher lifting force.

After the modification, the maximum takeoff weight of the helicopter is greater than 1000 grams. Thus the limitation on the avionics system is approximately 500 grams.

B. Avionics System

Weight and size of the on-board components are significant in the avionics system design, given the limited size and payload of the rotorcraft platform. On the other hand, performances of the system need to be guaranteed – processing speed and sampling rate of the components must be fast enough. A comprehensive survey is done on the state-of-the-art technologies, components are selected based primarily on their weight, size and performance, as well as availability and reliability. The avionics system consists of two on-board embedded microprocessors, an inertial measurement unit embedded with a 8 channels servo driver, an ultrasonic sonar range finder, a wireless module, and a camera.

In this paper, the avionics components will be classified into two platforms: The avionics control platform, and the vision processing platform.

C. Ground Supporting System

The ground supporting system incorporates a computer. It runs in a Linux operating system to give command and collect flight data from the on-board system for monitoring. Live video streaming from the camera on the UAV will be displayed in a customize software for observation purposes. Development of the control laws and the image processing algorithms of the UAV are also done in the ground station before transferring to the on-board microprocessors.

III. Avionics Control Platform

A. On-board Processor

The on-board embedded microprocessor is the most important components in the avionics system. It act as a brain of the whole system – collect flight data such as height and angular rates from the sensors, process the data, then send commands to the servo driver to execute appropriate control actions. Thus selecting suitable processors out of the available products in the industry is crucial to ensure the successful implementation of the UAV. Weight, size and performance are taken into account in choosing the on-board processors.

After much consideration, Gumstix Verdex Pro (Figure 4) is chosen to be the on-board processor for the control platform. Operating at a speed of 600MHz, Gumstix Verdex Pro together with the extension board, Gumstix console vx only weighs 23 grams. It has three Mini-Din 8 RS-232 ports for interfacing with peripheral components. In our design, they are used to interface with the IMU, the second on-board processor, and the modem for communication with ground station. Gumstix Buildroot built from Linux 2.6 kernel is installed as the operating system. It provides a cross-compilation tool-chain to compile and generate binary executable files which is compatible to the Gumstix processors. A 2GB Multimedia Card (MMC) is used to transfer files and to store flight data.

B. Inertial Measurement Unit

The MNAV100CA (Figure 5) is a digital sensor system integrated with servos controller. This module has a compact dimension of 5.72×4.57×2.54 mm with the weight of 33 grams. It contains all necessary sensors and drivers required to control a UAV despite its small size. The MNAV100CA module includes tri-axis accelerometers, tri-axis gyroscopes, tri-axis magnetometers and a GPS receiver module.

MNAV100CA is also embedded with a servo controller. It is able to drive up to 8 RC servos via PWM channels. It has a

Fig. 4. Gumstix Verdex Pro

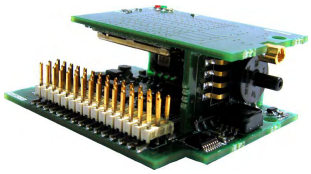

Fig. 5. MNAV100CA

Fig. 8. Failsafe Multiplexer

Fig. 9. Gumstix Overo Fire

Fig. 6. MaxSonar EZ4

Fig. 7. XBee-Pro Wireless Module

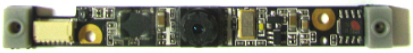

Fig. 10. Hp Pavilion DV5 Webcam

3-pin RS-232 serial port which allows the communication with the on-board processor. Performance wise, it has a switchable output rate of 50 or 100Hz which is sufficient to control our UAV.

In the absence of GPS signal in the indoor environment, the positions calculated by the IMU might not be accurate. Thus, vision based system is needed to correct the x- and y-axis positions, while a range finder is used to correct the z-axis measurement.

C. Ultrasonic Sonar Range Finder

The MaxSonar EZ4 (Figure 6) offers range detection and ranging in a small package of 19.9×21.1×16.4 mm and 4.3 grams. It is capable to detect objects and provides sonar range information ranging up to 645 centimeters with the sensitivity of 2.5 centimeters. It has three interface output formats – pulse width output, analog voltage output, and asynchronous serial digital output. To optimize the port usage of the on-board processor, analog voltage output is used for data transmission to the processor.

D. Wireless Communication Module

Wireless communication modules are needed to form communication link between the on-board system and the ground station. A pair of Xbee-Pro OEM RF modules is used to establish a data link between the on-board avionics system and the ground supporting system, while the build-in wireless module in Gumstix Overo is used to accomplish video link to the ground supporting system.

XBee-Pro wireless module (Figure 7) operates at 2.4GHz. Its indoor range of 100 meters and miniature size of 24.4×32.9 mm makes it an ideal wireless module to be implemented in the UAV. Besides, XBee-Pro can operate at the Transparent Mode, in which the module acts as a RS-232 serial cable replacement. The baud rate of the serial interface is set as 115200 bps in accordance with the Gumstix Verdex Pro computer console.

E. Safety Switch

In the event of an emergency, such as components break down or unstable control during the flight test, it is important that a human pilot to take back manual control quickly and easily [5]. A simple failsafe board (Figure 8) is used to toggle between automatic control and manual control. It has two 4 channels servo input ports and a 4 channels servo output port. This failsafe board operates like a relay, where there is an addition input channel to select which input port will be used as the output of the board. In our design, channel 5 of the receiver is used as the selector for this safety switch.

IV. Vision Processing Platform

A. On-board Vision Processor

To realize vision based navigation, a high performance microprocessor is needed to process images captured by the camera. The main processing unit for the embedded vision platform is Gumstix Overo Fire coupled with its Overo Summit expansion board (Figure 9). The attracting features of this unit include a 600 MHz processor, DSP coprocessor and Wifi connectivity. At the core of this system is a Texas Instruments OMAP3530 ARM processor, and is one of the fastest low powered embedded systems as of writing. This unit is small in physical dimensions and weighs 18 grams in total. In our design, the operating system provided by the manufacturer of Gumstix Overo has been replaced by a custom built version of the GNU/Linux operating system. Our system uses Embedded GLIBC (EGLIBC) as the core system library instead of the conventional GNU C Library (GLIBC) which is more suited for workstations. In addition, our custom built system provides a cleaner cross compiling environment for ease of software development.

A camera will be connected to Gumstix Overo via the mini USB port from the extension board. Data obtained from the vision processing algorithms will be sent to the control platform's processor via the UART port.

B. Camera

A camera is used to capture real time on-board video for image processing purpose. A webcam module taken from a Hp pavilion DV5 series laptop (Figure 10) is chosen to be

mounted in the UAV. This color CMOS camera has a tiny dimension of 8×80×6 mm and weight of 3 grams. Despite its small size, it is capable for providing 30 FPS at 320×240 resolutions. It is mounted below the platform of the UAV and is face vertically downwards to capture the colored tracks on the ground.

V. Vision Guided Control System

A. Image Processing Algorithm - Fast Path Detection

The innovation in this algorithm is to focus only on pixels near the border of the image frame. The assumption is that under normal conditions, if the path crosses the image frame, it must enter from one point and exit from another. By searching for the entry and exit points that lie on the border of the image, the direction and location of the path can be estimated.

The feature picked for detection is the edges between segments of different colors. Color classification was considered, but color classifiers are prone to misjudging colors under dynamic lighting conditions. Color edge on the other hand, are less prone to environmental changes.

A thin area is stripped from the sides of the image and resized to a single pixel height. The purpose of this two folds. Firstly, this allows the use of single dimension algorithms which are relatively cheap to conventional two-dimensional image processing algorithms. Secondly, resizing has the effect of averaging. Some noise is eliminated in this process.

Adaptive color edge detection is then used to split the stripped border into segments. As the image strip is single dimensioned, we avoid expensive convolution operations commonly used in 2D edge detection. Edge detection can be split into two steps in general. A difference function is first applied to evaluate the strength of a pixel as an edge, and a threshold is used to remove weak edges while preserving strong ones.

The function used to evaluate the strength of a pixel as an edge is the euclidean distance of two adjacent pixels in RGB color space. From our trails, making use of the three dimensional color space directly improves edge detection significantly as compared to conventional monochromatic methods. To avoid the problem of color bleeding, two nearby pixels are used in the computation and the maximum euclidean distance is selected as the edge strength of the pixel.

Given a pixel at position k with rgb value $\rho_k = \begin{pmatrix} r_k \\ g_k \\ b_k \end{pmatrix}$, it has an edge strength of d_k given by

$$d_k = \lfloor \max \left[|\rho_k - \rho_{k+1}|, |\rho_k - \rho_{k+2}| \right] \rfloor \tag{1}$$

Threshold values in edge detection are typically hand picked. Fixed threshold values are however, unable to adapt to changes in the environment. A reduction in ambiance lighting for example, may require a lower threshold value. Our group has chosen an adaptive technique that picks the optimal threshold value by maximizing entropy [9].

The goal of applying threshold is to divide pixels into two sets: pixels that form an edge ($d \geq T$), and pixels that do not form an edge ($d < T$). The total entropy of the two sets of pixels can be defined as

$$h(T) = H(d < T) + H(d \geq T) \tag{2}$$

Where H is the well known entropy measure

$$H(X) \triangleq -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{3}$$

The optimal threshold is found by maximizing h while varying T

$$\hat{T} = \max_T h(T) \tag{4}$$

Computation of $h(T)$ can be speed up by factoring common terms that are used in computing entropy of each set. Given the histogram, n of d, where n_i counts the occurrences of $d = i$,

$$\sigma_T = \sum_{i=1}^{T-1} n_i \tag{5}$$

$$\begin{aligned} H(d < T) &= -\sum_{i=1}^{T-1} \frac{n_i}{\sigma_T} \log_2 \frac{n_i}{\sigma_T} \\ &= \log_2 \sigma_T - \frac{1}{\sigma_T} \sum_{i=1}^{T-1} n_i \log_2 n_i \end{aligned} \tag{6}$$

The summation terms σ_T and $\sum_{i=1}^{T-1} n_i \log_2 n_i$ can be updated iteratively. A similar analysis can be done to obtain an iterative update for $H(d \geq T)$. The search for optimal $\hat{T}$ can thus be done in linear time.

After broken the image strip into segments of similar colors, the next step is to find the colors of each segment. Without the edge detection, we would have required a strong classifier that can place the color of each pixel accurately. Whereas with the knowledge of color segments, we require only a weak classifier to work on a group of pixels.

In this application, we require the classifier to recognize three basic colors, *Red*, *Green* and *Blue*. We have included an additional color for the case where the classifier has absolutely no confidence, for example when faced with a gray pixel. We term this "color" as *None*. Our color classifier is based on the concept of majority votes.

With segmentation by edge detection and color classification of each segment done, the task of detecting a path becomes trivial. The path can be found by searching for segments that run in the correct sequence of colors. This can be done in linear time with a simple finite state machine. The discovered segments that correspond to a path are then mapped back to the original image frame to resolve the coordinates of the mid point of upper and lower sections of the colored track.

Once the coordinates of these two points are obtained, the heading angle and the distance to the color path can be calculated as follows.

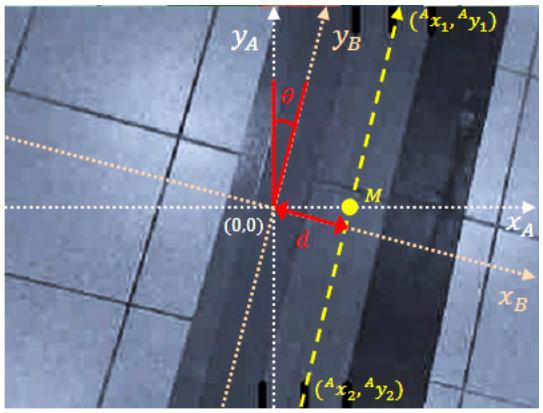

Fig. 11. Frames and coordinates system assigned to the same image shown in Figure 2

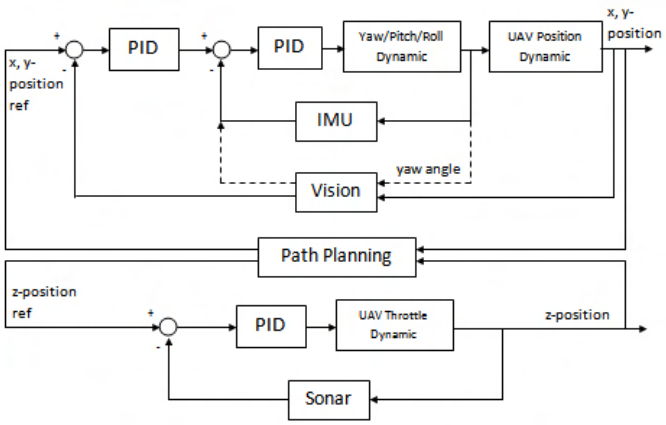

Fig. 12. The Overall Control Blocks Diagram

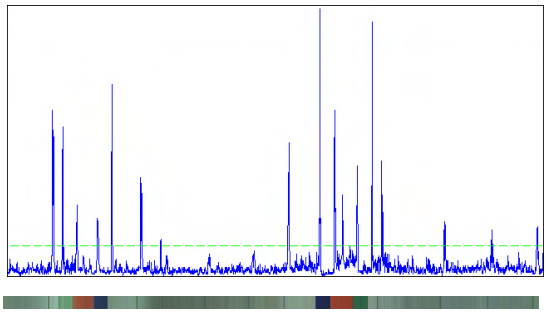

Fig. 13. Image strip and corresponding strength of each location as an edge. The horizontal dashed line denotes the optimal threshold.

Two frames are assigned to the system (Figure 11). Frame A represent the original frame of the UAV with x_A-axis pointed to the right and y_A-axis pointed to the front of the UAV. For simplicity, origin of frame B are assigned to be coincide with the origin of frame A, with y_B-axis parallel with the colored track.

Using these frame assignments, heading angle,

$$\theta = atan2({}^A x_1 - {}^A x_2, {}^A y_1 - {}^A y_2) \tag{7}$$

where positive angle corresponds to turning right and negative angle corresponds to turning left.

Coordinates of midpoint M in frame B,

$${}^B M = {}^B R_A \times M = {}^A R_B^{-1} \times M \equiv \begin{pmatrix} {}^B x \\ {}^B y \end{pmatrix} \tag{8}$$

where midpoint $M = \begin{pmatrix} \frac{{}^A x_1 + {}^A x_2}{2} & \frac{{}^A y_1 + {}^A y_2}{2} \end{pmatrix}^T$ and ${}^A R_B = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$ is the clockwise rotational matrix from frame A to frame B.

Referring to Figure 11, the distance to the colored tracks, d is the $x-$coordinate of point M in frame B, in this case, $d = {}^B x$.

B. Control Algorithm

Once the sensor measurements can be obtained accurately, an avionics control software can be implemented. An open source code package, MicroGear, is used to develop the control methodology. Our UAV is designed to fly in a near-hovering condition with minimal pitch and roll deviation during the navigation. Hence the yaw, pitch and roll angle of the UAV can be assumed to be decoupled to simplify the control tasks. These three angles will be control independently by individual controllers.

The flight control algorithm comprising three portions: the inner loop control, outer loop control as well as path planning. IMU sensors will provide feedback for the inner loop controller to control the stability of the UAV, such as the pitch and roll angles. Vision measurements provide feedback to the outer loop controller, which will control the heading angle and the position of the UAV. Beside the outer loop controller, path planning algorithm will generate position set point to drive the UAV. The overall control system of the UAV are summarized in the block diagram shown in Figure 12.

VI. Experimental Tests and Preliminary Results

A. Accuracy Tests

Few different tests are carried out to examine the accuracy of the sensors and to verify the vision processing and control algorithms of the UAV. Test results for each category will be presented in this section.

1) Path Detection: Adaptive color edge detection is applied to a thin strip around the captured image (Figure 2). Figure 13 shows the edge strength of each pixel, intersected by the optimal threshold computed. The algorithm efficiently edges from non-edges with only one false edge being produced due to imaging artifacts in this test.

2) Height Measurement: A simple experiment is carried out to verify the height measurement by the ultrasonic sonar sensor. It is done by manually oscillating the UAV position with respect to the ground. Upper part of Figure 14 reflects that the measurement result are generally accurate since there is no abrupt changes between any consecutive data point on the curve.

3) Lateral Position based on Vision: A similar experiment is done to verify the accuracy of vision based positioning. The

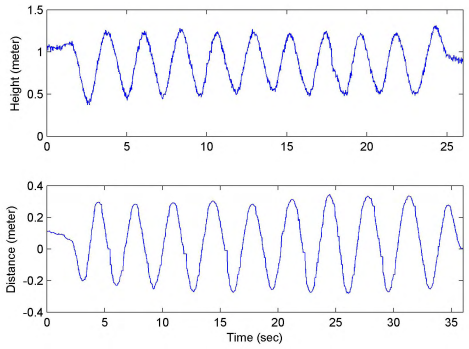

Fig. 14. **Up:** Height measured by the sonar range finder under oscillation motion. **Down:** Lateral position calculated by the vision algorithms under oscillation motion.

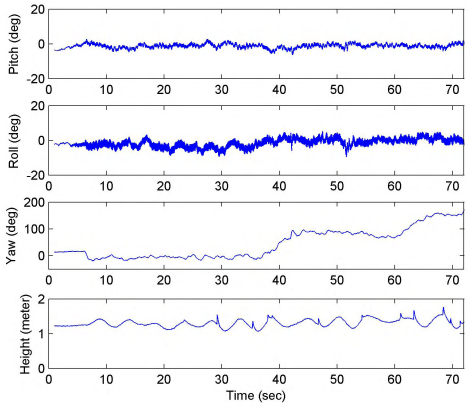

Fig. 15. One of the flight test results

UAV is manually oscillate laterally above the colored track, and the vision based calculated distance from the colored track is plotted as shown in lower part of Figure 14. The result generally shows that the real time lateral position calculated by our image processing algorithm is accurate without any unbearable noise. Note that the minor edges on the smooth sinusoidal curve are due to the failure of path detection at that instants. The previous stored value of lateral distance will be used when the path is not detected to preserve the stability of the UAV.

B. Real Time Flight Test

Real time flight tests are carried out in an indoor environment. In the fully autonomous mode, KingLion is taken off from hands and allowed to perform navigation by following the colored track on the ground. The Euler angles and the height of the UAV is logged for each flight. Figure 15 shows the height and Euler angles of the UAV in a single flight tests.

In this particular flight test, KingLion is commanded to navigate following the colored track at the height of 1.3 meter. Note from the *Yaw* plot, at time $t = 35$ sec and 60 sec, KingLion is performing a right turn. In overall, the height and Euler angles control of KingLion shows a reasonable performance. The complete flight test video is available at http://www.youtube.com/watch?v=CEkcA-_ji10.

VII. Conclusions

This paper summarized the development of our mini-UAV, KingLion, embedded with on-board vision processing as navigation system. From a simple bare RC helicopter, the avionics system is installed such that it is able to perform simple indoor navigation in autonomous mode. The avionics system includes necessary components such as microprocessors, IMU, wireless communication modules, camera etc. The vision processing and control algorithms such as fast path detection method are also discussed in detail. The performance of the UAV is also verified from the experiments and flight tests.

Some possible improvements to this project include implementing a more sophisticated inner loop control system such as non-linear control to optimize the performance of the UAV. As technologies advance, better sensors and actuators can be used to reduce the size and weight of the current UAV. In the future, the UAV is expected to perform more complicated indoor navigation and tasks such as obstacles avoidance and target tracking.

Acknowledgment

Authors of this paper would like to thank Dr. Guowei Cai and Mr. Fei Wang who have also contributed to the development of our UAV.

References

[1] Z. Harris, "Survey of uav applications in civil markets," in *Proceedings of the 9th Mediterrnean Conference on Control and Automation*, (Dubrovnik, Croatia), 2001.

[2] S. Bortoff, "The university of toronto rc helicopter: a testbed for nonlinear control," in *Proceedings of the 1999 IEEE Internatinoal Conference on Control Applications*, (Hawaii), pp. 333–338, 1999.

[3] M. Sugeno, I. Hirano, S. Nakamura, and S. Kotsu, "Development of an intelligent unmanned helicopter," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, (Yokohama, Japan), pp. 34–35, 2007.

[4] E. N. Sanchez, H. M. Becerra, and C. M. Velez, "Combining fuzzy, pid and regulation control for an autonomous mini-helicopter," in *Information Sciences 177 (2007)*, pp. 1999–2022, 2007.

[5] J. Roberts, P. Sikka, and L. Overs, "System design for a small autonomous helicopter," *Australian Conference on Robotics and Automation*, 2000.

[6] F. Wang, "Visual servo control for an unmanned aerial vehicle object tracking and uav positioning," tech. rep., National University of Singapore, 2008.

[7] A. R. Conway, *Autonomous Control of an Unstable Model Helicopter using Carrier Phase GPS only*. PhD thesis, Stanford University, March 1995.

[8] G. Cai, K. Peng, B. M. Chen, and T. H. Lee, "Design and assembling of a uav helicopter system," *The 5th International Conference on Control & Automation*, June 2005.

[9] R. Ji, B. Kong, F. Zheng, and J. Gao, "Color edge detection based on yuv space and minimal spanning tree," *International Conference on Information Acquisition*, August 2006.