A Robust Real-Time Embedded Vision System on an Unmanned Rotorcraft for Ground Target Following

Feng Lin, *Student Member, IEEE*, Xiangxu Dong, Ben M. Chen, *Fellow, IEEE*,
Kai-Yew Lum, *Member, IEEE*, and Tong H. Lee, *Member, IEEE*

***Abstract*—In this paper, we present the systematic design and implementation of a robust real-time embedded vision system for an unmanned rotorcraft for ground target following. The hardware construction of the vision system is presented, and an onboard software system is developed based on a multithread technique capable of coordinating multiple tasks. To realize the autonomous ground target following, a sophisticated feature-based vision algorithm is proposed by using an onboard color camera and navigation sensors. The vision feedback is integrated with the flight control system to guide the unmanned rotorcraft to follow a ground target in flight. The overall vision system has been tested in actual flight missions, and the results obtained show that the overall system is very robust and efficient.**

***Index Terms*—Image processing, real-time systems, target detection and following, unmanned aerial vehicles (UAVs), vision systems.**

I. Introduction

UNMANNED AERIAL VEHICLES (UAVs) have recently aroused much interest in the civil and industrial markets, ranging from industrial surveillance, agriculture, and academic research to wildlife conservation [6], [8], [14], [15], [26], [32], [34]. In particular, owing to its vertical takeoff-and-landing, hovering, and maneuvering capabilities, the unmanned rotorcraft has received much attention in the defense and security community [1]. More specifically, an unmanned rotorcraft equipped with a vision payload can perform a wide range of tasks, such as search and rescue, surveillance, target detection and tracking, etc., as vision provides a natural sensing modality—in terms of human comprehension—for feature detection and tracking [28], [29]. Instead of vision being merely a payload, many research efforts have also been devoted to vision-aided flight control [2], [17], [22], tracking [25], [28], terrain mapping [27], and navigation [18], [23].

We note that most of the works reported in the literature, however, focus on only a certain part of vision systems for UAVs, such as hardware construction or vision algorithms. Many of them are adopted from those designed for ground robots, which are not very suitable for applications on UAVs. To the best of our knowledge, there is hardly any systematic documentation in the open literatures dealing with the complete design and implementation of the vision system for unmanned rotorcrafts, which includes architectural and algorithmic designs of real-time vision systems. In addition, although the target tracking in video sequences has already been studied in a number of applications, there has been very little research related to the implementation of vision-based target following for UAVs.

In this paper, we present the design and implementation of a comprehensive real-time embedded vision system for an unmanned rotorcraft, which includes an onboard embedded hardware system, a real-time software system, and mission-based vision algorithms. More specifically, the onboard hardware system is designed to fulfill the image processing requirements by using the commercial off-the-shelf products, such as PC104 embedded modules. Real-time vision software is developed, which is running on the real-time operating system QNX. An advanced and efficient vision algorithm is then proposed and implemented to realize the ground target tracking, which is suited for the UAVs. The proposed vision scheme is integrated with the onboard navigation sensors to estimate the relative distance between the target and the UAV. Finally, using the vision feedback, a two-layer target tracking control framework is utilized to control a pan/tilt servomechanism to keep the target in the center of the image and guide the UAV to follow the motion of the target.

The remainder of this paper is organized as follows: Sections II and III present the development of hardware and software of the embedded vision system for a UAV, respectively, whereas coordinate systems adopted in the UAV vision systems are described in Section IV. Section V details the vision-based ground target detection and tracking algorithms, as well as the target-following scheme based on vision signal feedback. The experimental results of the vision system obtained through actual flight tests are presented in Section VI. Finally, we draw some concluding remarks in Section VII.

II. Hardware Configuration of the Vision System

The hardware configuration of the proposed onboard vision system for the UAV, as shown in Fig. 1, consists of the following five main parts: a visual sensor, an image acquisition module, a vision processing module, a pan/tilt servomechanism, and video and data links.

Manuscript received November 8, 2010; revised March 10, 2011; accepted May 19, 2011. Date of publication July 5, 2011; date of current version October 18, 2011.

F. Lin, X. Dong, B. M. Chen, and T. H. Lee are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 (e-mail: linfeng@nus.edu.sg; dong07@nus.edu.sg; bmchen@nus.edu.sg; eleleeth@nus.edu.sg).

K. Y. Lum is with Temasek Laboratories, National University of Singapore, Singapore 119260 (e-mail: tsllumky@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2011.2161248

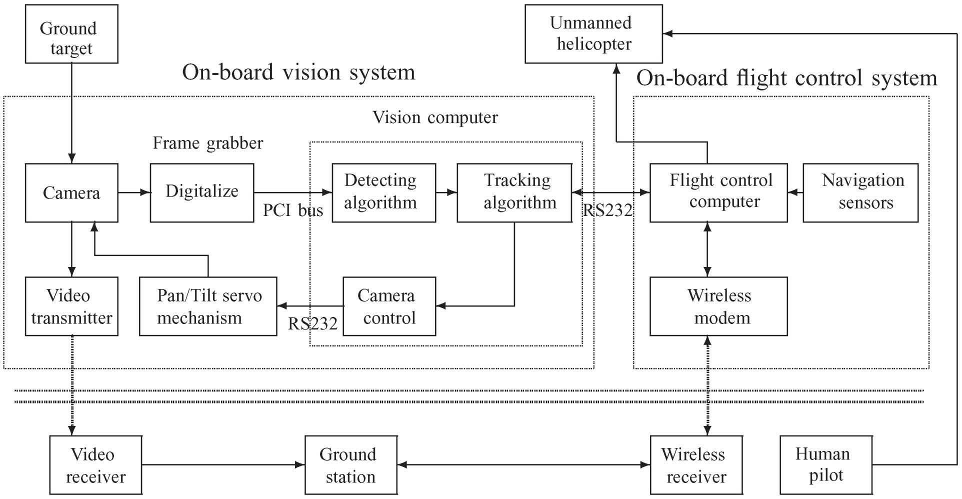


Fig. 1. Configuration of the overall vision system.

A. *Visual Sensor: Video Camera*

A visual sensor is employed on board to obtain in-flight visual information of the surrounding environment of the UAV. Interesting visual information is composed of silent and dynamic features, such as the color and shape of landmarks, and motions of vehicles. A color video camera is selected as the onboard visual sensor in our system, which has a compact size and a weight less than 30 g, as well as 380 TV line resolution and 40° field of view.

B. *Image Acquisition Module: Frame Grabber*

The primary function of a frame grabber is to perform the A/D conversion of the analog video signals and then output the digitalized data to a host computer for further processing. Our selection is a PC/104(-plus)-standard frame grabber, a Colory 104, which has the following features: 1) high resolution—Colory 104 is capable of providing a resolution of up to 720 × 576 (pixels), which is sufficient for online processing; 2) multiple video inputs—it is able to collect data from multiple cameras; 3) sufficient processing rate—the highest A/D conversion rate is 30 frames per second (FPS), which is higher than the onboard vision processing rate (10 FPS); and 4) featured processing method—two tasks are used alternatively to convert the digital video signal into specified formats.

C. *Vision Processing Module: Vision Computer*

As shown in Fig. 1, the digitalized visual signals provided by the frame grabber are transferred to the onboard vision computer that is the key unit of the vision system. The vision computer coordinates the overall vision system, such as image processing, target tracking, and communicating with the flight control computer, which is to be described in detail later in Section V. In this paper, the configuration of using two separated embedded computers in the onboard system for UAVs is proposed: one for flight control and another one for machine vision algorithms. We choose such a configuration for onboard system because of the following reasons: 1) the computation consumption of flight control task and vision program are very heavy, which can hardly be carried out together in a single embedded computer; 2) the sampling rate of the flight control computer is faster than the vision computer, since the faster sampling rate is required to stabilize the unmanned rotorcraft; 3) the decoupled structure reduces the negative effect of data blocking caused by the vision program and flight control system and thus makes the overall system more reliable.

In the proposed vision system, a separated onboard PC104 embedded computer, Cool RoadRunner III, is employed to process the digitalized video signal and execute the vision algorithms. The core of the board is an Intel LV Pentium-III processor running at 933 MHz. A compact flash memory card is used to save the captured images.

D. *Pan/Tilt Servomechanism*

In the application of the ground target following, it is required to keep the target objects in the field of view of the camera to increase the flexibility of vision-based tracking. As such, we decide to mount the camera on a pan/tilt servomechanism that can rotate in the horizontal and vertical directions.

E. *Wireless Data Link and Video Link*

In order to provide ground operators with clear visualization to monitor the work that the onboard vision is processing during flight tests, the video captured by the onboard camera is transmitted and displayed in a ground control station. An airborne 2.4-GHz wireless video link is used to transmit the live video captured to the ground control station.

III. Configuration of the Vision Software System

Based on the proposed hardware system, the configuration of the onboard vision software system is presented. The purpose of the vision software system is to coordinate the work of onboard devices and implement vision algorithms. Since the vision software system targets for real-time applications and runs in an embedded PC104 computer, QNX Neutrino, a real-time embedded operating system is employed as the developing platform. QNX Neutrino has a microkernal that requires fewer system resources and performs more reliably and efficiently for embedded systems during runtime compared to the traditional monolithic kernel.

The vision software program coordinates tasks such as capturing video, controlling pan/tilt servomechanism, and performing the vision detecting and tracking algorithms. To make the vision software system easy to design and robust to perform, the entire vision software system is divided into several main blocks. Each block is assigned a special task as follows.

1) CAM: Reading RGB24 images from the buffers assigned to the frame grabber. The reading rate is set up to be 10 FPS. In order to reduce the risk of damaging the image data, two buffers are used to store the captured images by the frame grabber alternatively.
2) IMG: Processing the captured images and carrying out the vision algorithms, such as the automatic tracking and camera control, which will be explained in Section V.
3) SVO: Controlling the rotation of the pan/tilt servomechanism to keep the ground target in a certain location of the image.
4) SAV: Saving the captured and processed images to a high-speed compact flash.
5) COM: Communicating with the flight control computer. The flight control computer sends the states of the UAV and commands from the ground station to the vision computer, and the vision computer sends the estimated relative distance between the UAV and the ground target to the flight control computer to guide the flight of the UAV.
6) USER: Providing a mean for users to control the vision program such as running and stopping the tracking, as well as changing the parameters of the vision algorithms.
7) MAIN: Managing and scheduling the work of the entire vision software system.

IV. Coordinate Frames Used in Vision Systems

Shown in Fig. 2 are the coordinate systems adopted in the UAV vision systems. More specifically, we have the following.

1) Local north–east–down (NED) coordinate system (labeled with a subscript "n") is an orthogonal frame on the surface of the Earth, whose origin is the launching point of the aircraft on the surface of the Earth.
2) Body coordinate system (labeled with subscript "b") is aligned with the shape of the fuselage of the aircraft.
3) Servo base coordinate system (labeled with subscript "s") is attached to the base of the pan/tilt servomechanism, which is aligned with the body coordinate system of the UAV.
4) Spherical coordinate system (labeled with subscript "sp") is also attached to the base of the pan/tilt servomechanism. It is used to define the orientation of the camera and the target with respect to the UAV. Given a generic point $\mathbf{p}_s = (x_s, y_s, z_s)^{\mathrm{T}}$ in the servo base coordinate system, its position can be defined in the spherical coordinate system by three numbers: radius r_{sp}, azimuth angle θ_{sp}, and elevation angle ϕ_{sp}, which is given by

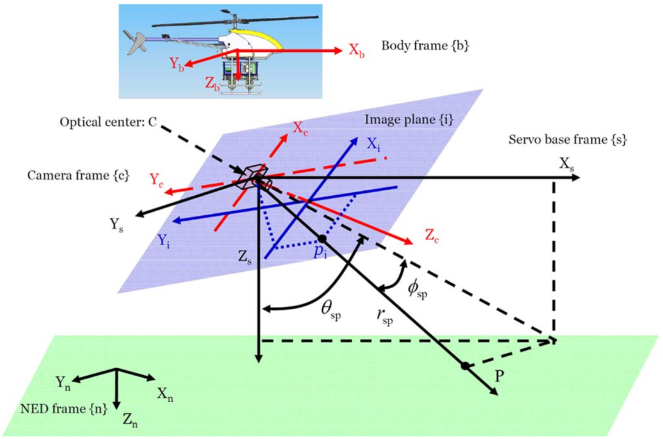


Fig. 2. Coordinate frames used in unmanned vision systems.

$$\mathbf{p}_{\mathrm{sp}} = \begin{pmatrix} r_{\mathrm{sp}} \\ \theta_{\mathrm{sp}} \\ \phi_{\mathrm{sp}} \end{pmatrix} = \begin{pmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \tan^{-1}\left(\frac{x_s}{z_s}\right) \\ \sin^{-1}\left(\frac{y_s}{r_{\mathrm{sp}}}\right) \end{pmatrix}. \tag{1}$$

5) Camera coordinate system (labeled with subscript "c"), whose origin is the optical center of the camera. The Z_c-axis is aligned with the optical axis of the camera and points from the optical center C toward the image plane.
6) Image frame (or the principle image coordinate system) (appended with subscript "i") has the origin at the principal point. The coordinate axes X_i and Y_i are aligned with the camera coordinate axes X_c and Y_c, respectively.

V. Vision-Based Ground Target Following

To realize the vision-based ground target detection, many vision approaches have been proposed worldwide, such as template matching [7], [28], background subtraction [19], [35], optical flow [3], [18], stereo-vision-based technologies [11], and feature-based approaches [20], [31], [36], [39].

In this paper, a sophisticated vision-based target detection and tracking scheme is proposed, as shown in Fig. 3, which employs robust feature descriptors and efficient image-tracking techniques. Based on the vision sensing data and navigation sensors, the relative distance to the target is estimated. Such estimation is integrated with the flight control system to guide the UAV to follow the ground target in flight.

A. Target Detection

The purpose of the target detection is to identify the target of interest from the image automatically based on a database of preselected targets. A toy car is chosen as the ground target. A classical pattern recognition procedure is used to identify the target automatically, which includes three main steps, i.e., segmentation, feature extraction, and pattern recognition.

1) Segmentation: The segmentation step aims to separate the objects of interest from background. To simplify the further processing, some assumptions are made. First, the target

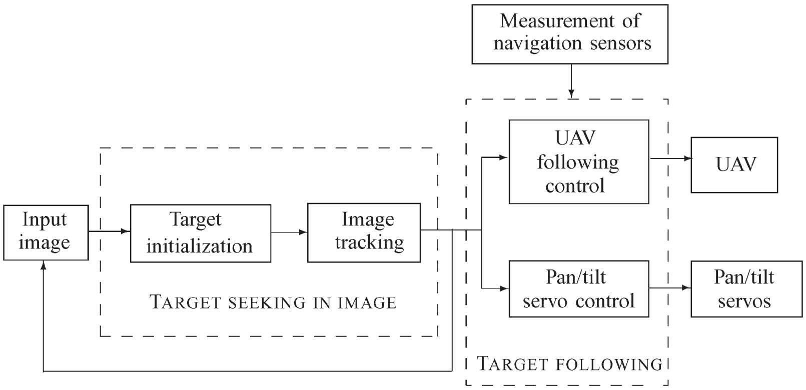


Fig. 3. Flow chart of the ground target detection, tracking, and following.

and environments exhibit *Lambertian reflectance*, and in other words, their brightness is unchanged regardless of viewing directions. Second, the target has a distinct color distribution compared to the surrounding environments.

Step 1) *Threshold in color space*. To make the surface color of the target constant and stable under the varying lighting condition, the color image is represented in HSV space, which stands for hue (hue), saturation (sat), and value (val) introduced originally by Smith [33]. Precalculated threshold ranges are applied to the hue, sat, and val channels

$$hue_r = [h_1,\ h_2] \qquad sat_r = [s_1,\ s_2] \qquad val_r = [v_1,\ v_2]. \tag{2}$$

Only the pixel values falling in these color ranges are described as the foreground points, and pixels of the image that fall out of the specified color range are removed. The procedure of the image preprocess is shown in Fig. 4.

Step 2) *Morphological operation*. As shown in Fig. 4, normally, the segmented image is not smooth and has many noise points. Morphological operations are then employed to filter out noise, fuse narrow breaks and gulfs, eliminate small holes, and fill gaps in the contours. Next, a contour detection approach is used to obtain the complete boundary of the objects in the image, which will be used in the feature extraction.

2) Feature Extraction: Generally, multiple objects will be found in the segmented images, including the true target and false objects. The geometric and color features are used as the descriptors to identify the true target.

Geometry Feature Extraction: To describe the geometric features of the objects, the four lowest moment invariants proposed in [25] are employed since they are independent of position, size, and orientation in the visual field. The four lowest moment invariants, defined in the segmented image $I(x, y)$, are given by

$$\phi_1 = \eta_{20}^m + \eta_{02}^m \tag{3}$$

$$\phi_2 = \left(\eta_{20}^m - \eta_{02}^m\right)^2 + 4\left(\eta_{11}^m\right)^2 \tag{4}$$

$$\phi_3 = \left(\eta_{30}^m - 3\eta_{12}^m\right)^2 + \left(\eta_{03}^m - 3\eta_{21}^m\right)^2 \tag{5}$$

$$\phi_4 = \left(\eta_{30}^m + \eta_{12}^m\right)^2 + \left(\eta_{03}^m + \eta_{21}^m\right)^2 \tag{6}$$

where η_{pq}^m, for $p + q = 2, 3, \ldots$, is the improved normalized central moment defined as

$$\eta_{pq}^m = \frac{\mu_{pq}^c}{A^{(p+q+1)/2}} \tag{7}$$

where A is the interior area of the shape and μ_{pq}^c is the central moment defined as

$$\mu_{pq}^c = \int\limits_C (x - \bar{x})^p (y - \bar{y})^q \, ds, \qquad p, q = 0, 1, \ldots. \tag{8}$$

Note that, in (8), C is the boundary curve of the shape, $\int_C$ is a line integral along C, $ds = \sqrt{(dx)^2 + (dy)^2}$, and $[\bar{x}, \bar{y}]$ is the coordinate of the centroid of the shape in the image plane.

In addition, compactness is another useful feature descriptor for recognition. Compactness of a shape is measured by the ratio of the square root of the area and the perimeter, which is given by

$$Compactness: \beta_c = \frac{\sqrt{A}}{C}. \tag{9}$$

It can be easily proven that compactness is invariant with respect to translation, scaling, and rotation.

Color Feature Extraction: To make the target detection and tracking more robust, we also employ color histogram to represent the color distribution of image area of the target, which is not only independent of the target orientation, position and size but also robust to partial occlusion of the target and easy to implement. Due to the stability in outdoor environments, only hue and val are employed to construct the color histogram for object recognition, which is defined as

$$H = \{hist(i, j)\}, \qquad i = 1, \ldots, N_{\text{hue}}; \quad j = 1, \ldots, N_{\text{val}} \tag{10}$$

where

$$hist(i, j) = \sum_{(x,y)\in\Omega} \delta\left(i, \left[\frac{hue(x, y)}{N_{\text{hue}}}\right]\right) \delta\left(j, \left[\frac{val(x, y)}{N_{\text{val}}}\right]\right)$$

and where N_{hue} and N_{val} are the partition numbers of hue and val channels, respectively, Ω is the region of the target, $[\cdot]$ is the nearest integer operator, and $\delta(a, b)$ is the Kronecker delta function.

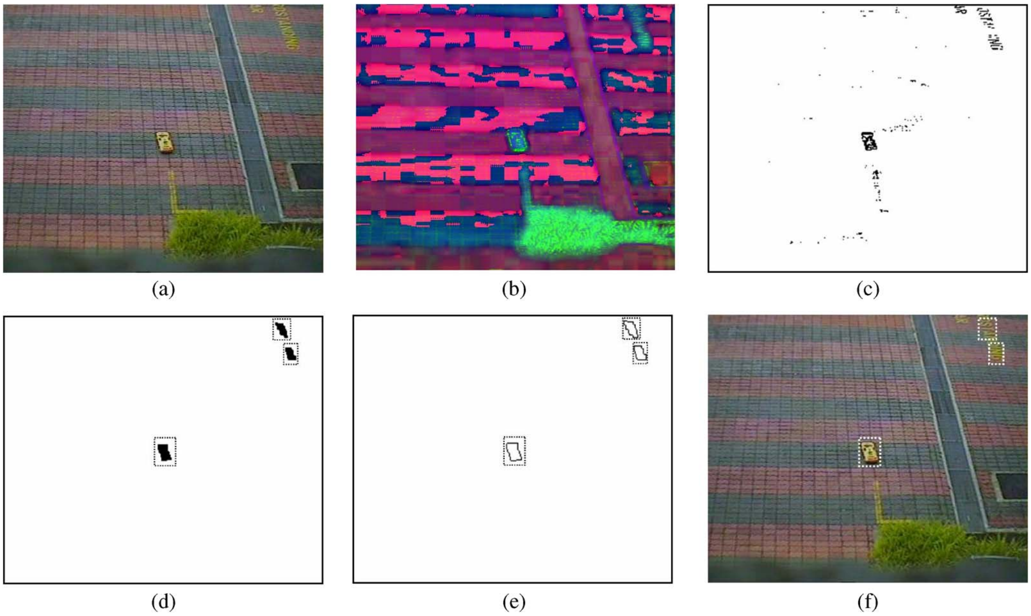

(a) (b) (c)
(d) (e) (f)

Fig. 4. Illustration of segmentation. (a) Input image. (b) Image in HSV color space. (c) Image after thresholding. (d) Image after morphological operations. (e) Image after contour detection. (f) Regions of interest.

Dynamic Features: Aside from the static features extracted from the foreground objects, we further calculate their dynamic motion using the Kalman filtering technique. The distance between the location of each object z_i and the predicted location of the target $\hat{z}$ is employed as a dynamic feature. The detailed procedure for predicting the location of the target in the image is to be discussed in Section V-B1. Both the static and dynamic features of them are then employed in the pattern recognition.

The extracted features of an object need to be arranged in a compact and identifiable form [30]. A straightforward way is to convert these features in a high-dimensional vector. For example, the feature vector of the ith object is given by

$$\begin{aligned}\alpha_i &= [\beta_{c,i},\ \phi_{1,i},\ \phi_{2,i},\ \phi_{3,i},\ \phi_{4,i},\ H_i,\ \mathbf{z}_i] \\ &= \{\alpha_{k,i}\}, \qquad k = 1, \ldots, d \end{aligned} \tag{11}$$

where d is the dimension of the feature vector.

3) Pattern Recognition: The purpose of the pattern recognition is to identify the target from the extracted foreground objects in terms of the extracted features in (11). The straightforward classifier is to use the nearest neighbor rule. It calculates a metric or "distance" between an object and a template in a feature space and assigns the object to the class with the highest scope. However, to take advantage of *a priori* knowledge of the feature distribution, the classification problem is formulated under the model-based framework and solved by using a probabilistic classifier. A discriminant function, derived from Bayes' theorem, is employed to identify the target. This function is computed based on the measured feature values of each object and the known distribution of features obtained from training data.

Step 1) *Prefilter*: Before classifying the objects, a prefilter is carried out to remove the objects whose feature values are outside certain regions determined by *a priori* knowledge. This step aims to improve the robustness of the pattern recognition and speed up the calculation.

Step 2) *Discriminant function*: We use the discriminant function, derived from Bayes' theorem, to determine the target based on the measured feature values of each object and the known distribution of features of the target obtained from the training data. We assume that these features are independent and fulfill normal distributions. Thus, we can define the simplified discriminant function with weightings as

$$f'_j(\alpha_i) = \sum_{k=1}^{5} w_k \left(\frac{\alpha_{k,i} - \mu_{k,j}}{\sigma_{k,j}}\right)^2 + w_6 \left(\frac{d_c(H_i, G_j) - \mu_{6,j}}{\sigma_{6,j}}\right)^2$$

where

$$d_c(H_i, G_j) = \frac{\sum_{p=1}^{N_h} \sum_{q=1}^{N_v} \min\left(H_i(p,q), G_i(p,q)\right)}{\min\left(|H_i|, |G_j|\right)} \tag{12}$$

and $\alpha_{k,i}$ is the kth element of the feature vector of the object i. $\mu_{k,j}$ and $\sigma_{k,j}$ are the mean and standard deviation of the distribution of the corresponding feature. G_j is the color histogram template of a predefined target. In fact, the location information is not used in the detection mode. The target i with the minimum value is considered as the candidate target. w_1 to w_6 are the weighting scalars of the corresponding features. In terms of the likelihood values of the objects, a decision rule is defined as

$$D = \begin{cases} target = \arg\min_i f'_j(\alpha_i), & \text{which belongs to class } j, \\ & \text{if } \min f'_j(\alpha_i) \le \Gamma'_j \\ \text{no target in the image,} & \\ & \text{if } \min f'_j(\alpha_i) > \Gamma'_j \end{cases}$$

where Γ' is a threshold value chosen based on the training data. This decision rule chooses the object, for example, i, with the smallest value of the simplified discriminant function as the candidate target. If $f'_j(\alpha_i) < \Gamma'_j$, then the scheme decides that object i is the target. Otherwise, the scheme indicates that there is no target in the current image.

B. Image Tracking

As shown in Fig. 3, after initialization, the image-tracking techniques are employed. The purpose of image tracking is to find the corresponding region or point to the given target. Unlike the detection, the entire image search is not required. Thus, the processing speed of image tracking is faster than the detection. The image-tracking problem can be solved by using two main approaches: 1) filtering and data association and 2) target representation and localization [13].

Filtering and Data Association: The filtering and data association approach can be considered as a top–down process. The purpose of the filtering is to estimate the states of the target, such as static appearance and location. Typically, the state estimation is achieved by using filtering technologies [38], [40]. It is known (see, for example, [24]) that most of the tracking algorithms are model based because a good model-based tracking algorithm will greatly outperform any model-free tracking algorithm if the underlying model is found to be a good one. If the measurement noise satisfied the Gaussian distribution, the optimal solution can be achieved by the Kalman filtering technique [4]. In some more general cases, particle filters are more suitable and robust [21]. However, the computational cost increases, and the sample degeneracy is also a problem. When multiple targets are tracked in the image sequence, the validation and association of the measurements become a critical issue. The association techniques, such as probabilistic data association filter (PDAF) and joint PDAF are widely used [37].

Target Representation and Localization: Aside from using the motion prediction to find the corresponding region or point, the target representation and localization approach is considered as another efficient way, which is referred to as a bottom–up approach. Among the searching methods, the mean-shift approach using the density gradient is commonly used [5], which is trying to search the peak value of the object probability density. However, the efficiency will be limited when the spatial movement of the target becomes significant.

To take advantages of the aforementioned approaches, using multiple trackers is widely adopted in applications of image tracking. In [37], the tracking scheme by integrating motion, color, and geometric features was proposed to realize robust image tracking. In conclusion, combining the motion filtering and advanced searching algorithms will definitely make the tracking processing more robust, but the computational load is heavier.

In our approach, instead of using multiple trackers simultaneously, a hierarchical tracking scheme is proposed to balance the computational cost and performance, which is shown in Fig. 5. In the model-based image tracking, the Kalman filtering technique is employed to provide accurate estimation and prediction of the position and velocity of a single target, referred to as dynamic information. If the model-based tracker fails to find the target, a mean-shift-based image-tracking method will be activated to retrieve the target back in the image.

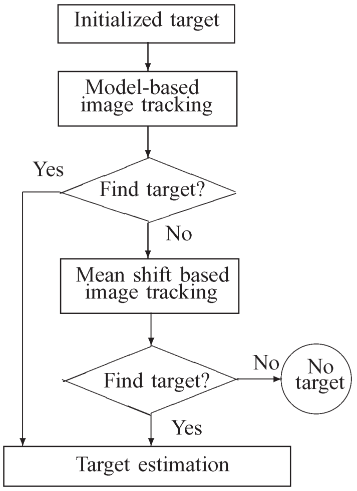


Fig. 5. Flow chart of image tracking.

1) Model-Based Image Tracking: Model-based image tracking will predict the possible location of the target in the subsequent frames and then do the data association based on an updated likelihood function. The advantage of the model-based image tracking is to combine dynamic features with geometric features of the target in the image tracking under noise and occlusion condition. In addition, several methods are employed to make the tracking more robust and efficient, which are given by the following:

1) narrow the search window in terms of the prediction of the Kalman filter;
2) integrate the spatial information with appearance and set the different weightings for the discriminant function.

The motion of the centroid of the target $\mathbf{x} = [\bar{x}, \dot{\bar{x}}, \bar{y}, \dot{\bar{y}}]^{\mathrm{T}}$ in the 2-D image coordinate is tracked using a fourth-order Kalman filter, which predicts the possible location of the target in the successive frames. The discrete-time model of the target motion can be expressed as

$$\begin{aligned} \mathbf{x}(k|k-1) &= \mathbf{\Phi}\mathbf{x}(k-1) + \mathbf{\Lambda}\mathbf{w}(k-1) \\ \mathbf{z}(k) &= \mathbf{H}x(k) + \mathbf{v}(k) \end{aligned} \tag{13}$$

where $\mathbf{w}$ and $\mathbf{v}$ denote the input and measurement zero-mean Gaussian noises

$$\Phi = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \Lambda = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

where T_s is the sampling period of the vision-based tracking system. A Kalman filter can then be designed based on the aforementioned motion model to estimate the states of the target in the image plane. The filter consists of the following stages.

1) Predicted state

$$\hat{\mathbf{x}}(k|k-1) = \Phi\hat{\mathbf{x}}(k-1).$$

2) Updated state estimate

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)\left(\mathbf{z}(k) - H\hat{\mathbf{x}}(k|k-1)\right)$$

where $\mathbf{K}(k)$ is the optimal Kalman gain.

The distance between the location of each object $\mathbf{z}_i$ and the predicted location of the target $\hat{\mathbf{z}}$ is employed as the dynamic feature defined by

$$\tilde{\mathbf{z}}_i = \mathbf{z}_i(k) - \hat{\mathbf{z}}(k) = \mathbf{z}_i(k) - H\hat{\mathbf{x}}(k|k-1).$$

Thus, the updated discriminant function, which includes the appearance and spatial information, is shown as follows:

$$f'_j(\alpha_i) = \sum_{k=1}^{5} w_k \left(\frac{\alpha_i(k) - \mu_j(k)}{\sigma_j(k)}\right)^2 + w_6\left(\frac{d_c(H_i, G_j) - \mu_j(6)}{\sigma_j(6)}\right)^2 + w_7\left(\frac{\|\tilde{\mathbf{z}}_i\| - \mu_j(7)}{\sigma_j(7)}\right)^2. \quad (14)$$

Most of the time, the model-based tracker can lock the target in the image sequence, but sometimes, it may fail due to the noise or disturbance, such as partial occlusion. Thus, a scheme is required to check whether the target is still in the image and then activate other trackers.

2) Switching Mechanism: The purpose of the switching mechanism is to check whether the target is still in the image when the target is lost by the model-based tracker. If yes, the mean-shift tracker will be activated. The lost of the target can be attributed to the poor match of features due to noise, distortion, or occlusion in the image. An alternative reason may be the maneuvering motion of the target, and the target is out of the image. Therefore, in order to know the reason and take the special way to find the target again, it is necessary to formulate the decision making as the following hypothesis testing problem:

H_0 : The target is still in the image;

H_1 : The target is not in the image due to maneuvers.

The estimation error is considered as a random variable, which is defined by

$$\varepsilon = (\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{z}_{k-1})^T \Sigma^{-1} (\mathbf{H}\hat{x}_{k-1} - z_{k-1})$$

where $\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{z}_{k-1}$ is assumed to be $N(0, \Sigma)$ distributed. ε is Chi-square distributed with two degrees of freedom (x and y directions) under H_0

$$\begin{cases} \varepsilon < \lambda = \chi_2^2(\alpha), & \text{if } H_0 \text{ is true} \\ \varepsilon \geq \lambda = \chi_2^2(\alpha), & \text{if } H_1 \text{ is true} \end{cases}$$

where $1 - \alpha$ is the level of confidence, which should be sufficiently high (for our system, $1 - \alpha = 99\%$). If H_0 is true, the Chi-square testing-based switching declares that the target is still in the image and enables the mean-shift-based tracker.

3) Mean-Shift-Based Image Tracking: If the target is still in the image, continuously adaptive mean-shift (CAMSHIFT) algorithm [5] is employed, which is shown in Fig. 5. This algorithm uses the mean-shift searching method to efficiently obtain the optimal location of the target in the search window. The principle idea is to search the dominated peak in the feature space based on the previous information and certain assumptions. The detected target is verified by comparing with an adaptive target template. The CAMSHIFT algorithm consists of three main steps: back projection, mean-shift searching, and search window adaptation.

Step 1) *Back projection*: In order to search the target in the image, the probability distribution image needs to be constructed based on the color distribution of the target. The color distribution of the target defined in hue channel is given by

$$hist_{\mathrm{tg}}(i) = \sum_{(x,y)\in\Omega} \delta\left(i, \left[\frac{hue_{\mathrm{tg}}(x,y)}{N_{\mathrm{hue}}}\right]\right), \qquad i = 1, \ldots, N_{\mathrm{hue}}.$$

Based on the color model of the target, the back projection algorithm is employed to convert the color image to the color probability distribution image. The probability of each pixel $\mathbf{I}_p(x,y)$ in the region of interest Ω_r is calculated based on the model of the target, which is used to map the histogram results and given by

$$\mathbf{I}_p(x,y) = hist_{\mathrm{tg}}\left(\left[\frac{\mathbf{I}_{\mathrm{hue}}(x,y)}{N_{\mathrm{hue}}}\right]\right) \quad (15)$$

where $\mathbf{I}_{\mathrm{hue}}$ is the pixel values of the image in the hue channel.

Step 2) *Mean-shift algorithm*: Based on the obtained color density image, a robust nonparametric method, the mean-shift algorithm, is used to search the dominated peak in the feature space. The mean-shift algorithm is an elegant way of identifying these locations without estimating the underlying probability density function [12].

Recalling the discrete 2-D image probability distributions in (15), the mean location (the centroid) of the search window is computed by

$$x_c(k) = \frac{\mathbf{M}_{10}}{\mathbf{M}_{00}} \qquad y_c(k) = \frac{\mathbf{M}_{01}}{\mathbf{M}_{00}}$$

where k is the number of iterations

$$\mathbf{M}_{00} = \sum_{(x,y)\in\Omega_w} \mathbf{I}_p(x,y)$$

$$\mathbf{M}_{10} = \sum_{(x,y)\in\Omega_w} \mathbf{I}_p(x,y)x \qquad \mathbf{M}_{01} = \sum_{(x,y)\in\Omega_w} \mathbf{I}_p(x,y)y$$

where Ω_w is the region of the search window, $\mathbf{M}_{00}$ is the zeroth moment, and $\mathbf{M}_{10}$ and $\mathbf{M}_{01}$ are the first moments for x and y, respectively. The search window is centered at the mean location $\mathbf{c}(k) = (x_c(k), y_c(k))$. Step 2) is to be repeated until $\|\mathbf{c}(k) - \mathbf{c}(k-1)\| < \varepsilon$.

Step 3) *Search window adaptation*: The region of interest is calculated dynamically using the motion filtering given in Section V-B1. To improve the performance of the CAMSHIFT algorithm, multiple search windows in the region of interest are employed. The initial locations and sizes of the searching windows are adopted from the centers and boundaries of the foreground objects, respectively. These foreground objects are obtained using the color segmentation in the region of interest. In the CAMSHIFT algorithm, the size of the search window will be dynamically updated according to the moments of the region inside the search window [5]. Generally, more than one target candidate will be detected due to multiple search windows adopted. To identify the true target, the similarity between the target model and the detected target candidate is measured using the intersection comparison (12). This verification can effectively reduce the risk of detecting the false target.

C. Target-Following Control

We proceed to design a comprehensive target-following system in this section. It consists of two main layers: the pan/tilt servomechanism control and the UAV following control. The overall structure of the target-following control is shown in Fig. 6. As mentioned in Section II, a pan/tilt servomechanism is employed in the first layer to control the orientation of the camera to keep the target in an optimal location in the image plane, namely, eye-in-hand visual servoing [9], [10], which makes target tracking in the video sequence more robust and efficient. The parameters associated with the pan/tilt servo control in Fig. 6 are to be introduced in detail later. In the second layer, the UAV is controlled to maintain a constant relative distance between the moving target and the UAV in flight.

1) Control of the Pan/Tilt Servomechanism: As shown in Fig. 6, given a generic point $\mathbf{P}$, $\mathbf{p}_i$ and $\mathbf{p}_i^*$ are the measured and desired locations of the projected point $\mathbf{P}$ in the image plane, respectively. $\mathbf{e} = [e_\phi, e_\theta]^{\mathrm{T}}$ is the tracking error, $\mathbf{u} = [u_\phi, u_\theta]^{\mathrm{T}}$ is the output of the tracking controller, and $\mathbf{v} = [v_\phi, v_\theta]^{\mathrm{T}}$ is the output of the pan/tilt servomechanism. M is the camera model, which maps the points in the 3-D space to the projected points in the 2-D image frame. N is a function to calculate the orientation of an image point $\mathbf{p}_i$ with respect to the UAV under the current $\mathbf{v}$. As mentioned in the definitions of the coordinate systems, the orientation of $\mathbf{P}$ with respect to the UAV can be defined using azimuth and elevation angles in the spherical coordinate system, which is described by two rotation angles $\mathbf{p}_e = [p_\phi, p_\theta]^{\mathrm{T}}$.

In image processing, the distortion of the lens is compensated, and the origin of the image plane is set as the principal point. Thus, we can obtain a simplified pinhole projection model as

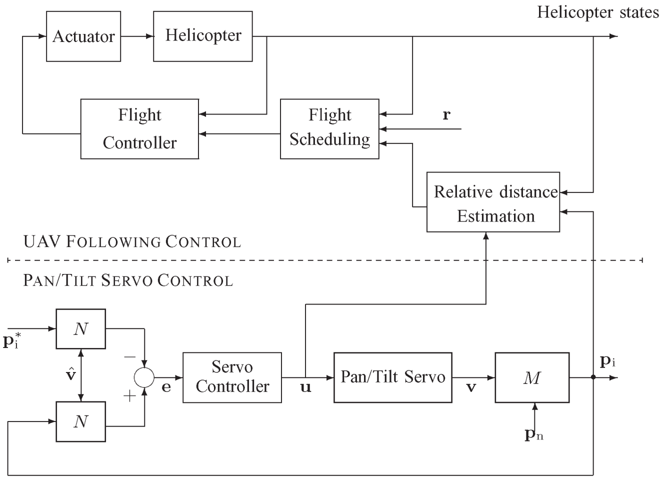


Fig. 6. Block diagram of the tracking control scheme.

$$\begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} = \frac{1}{\lambda} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_c \tag{16}$$

with

$$\mathbf{p}_c = \mathbf{R}_{c/n}\mathbf{p}_n + \mathbf{t}_{c/n} \tag{17}$$

where $\lambda = z_c$ is the depth of the point $\mathbf{P}$ in the camera coordinate system; f_x and f_y are the vertical and horizontal focal lengths in pixels, respectively; and $\mathbf{R}_{c/n}$ and $\mathbf{t}_{c/n}$ are the rotation matrix and the translation vector, respectively, which define the rigid-body transformation from the NED frame to the camera frame. Thus, we can define M as

$$\mathbf{p}_i = M(\mathbf{p}_n, \mathbf{v}) = \frac{1}{\lambda} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \end{bmatrix} \mathbf{R}_{c/n}(\mathbf{p}_n - \mathbf{t}_{c/n}).$$

Next, to derive the function N, we write the transformation between the camera coordinate system and the servo base coordinate system as

$$\mathbf{p}_s = \mathbf{R}_{s/c}(\mathbf{v})\mathbf{p}_c \tag{18}$$

where $\mathbf{p}_s$ is the coordinate of the point $\mathbf{P}$ relative to the servo base coordinate system and $\mathbf{R}_{s/c}$ describes the rotation from the servo base frame to the camera frame. We can then combine (18) with (16) and define the coordinate of the target in the spherical coordinate system

$$\mathbf{p}_e = \begin{pmatrix} p_\phi \\ p_\theta \end{pmatrix} = N(\mathbf{p}_i, \mathbf{v}) = \begin{pmatrix} \sin^{-1}\left(\frac{\bar{y}_s}{\bar{r}_{\mathrm{sp}}}\right) \\ \tan^{-1}\left(\frac{\bar{x}_s}{\bar{z}_s}\right) \end{pmatrix} \tag{19}$$

where

$$\begin{pmatrix} \bar{x}_s \\ \bar{y}_s \\ \bar{z}_s \end{pmatrix} = \mathbf{R}_{s/c}(\mathbf{v}) \begin{bmatrix} f_x^{-1} & 0 & 0 \\ 0 & f_y^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$
$$\bar{r}_{\mathrm{sp}} = \sqrt{\bar{x}_s^2 + \bar{y}_s^2 + \bar{z}_s^2}. \tag{20}$$

The pan/tilt servomechanism can be approximately considered as two decoupled servomotors, which regulate the visual sensor for horizontal and vertical rotations, respectively. The dynamic model of the servomotor can be described by using a standard second-order system. Before proceeding to design the control law for the pan/tilt servomechanism, we define the tracking error function as

$$\mathbf{e}(k) = \mathbf{p}_e - \mathbf{p}_e^* = N\left(\mathbf{p}_i(k), \mathbf{v}(k)\right) - N\left(\mathbf{p}_i^*, \mathbf{v}(k)\right) \tag{21}$$

where $\mathbf{p}_e^*$ denotes the desired orientation of the camera. The control inputs will be sent to the pan/tilt servos after the vision-based target detection algorithm, which generally cost about one sampling period. To track the moving target efficiently, we calculate the pan/tilt servo control inputs using the predicted location of the target in the subsequent frame, which is derived from (13) and given by

$$\hat{\mathbf{p}}_i(k+1) = \hat{\mathbf{z}}(k+1|k) = H\hat{\mathbf{x}}(k+1|k). \tag{22}$$

In implementation, it is not easy to measure the output of the pan/tilt servo $\mathbf{v}$ in (21). We assume that the bandwidth of the pan/tilt servomechanism is much faster than that of the control system. We then can ignore the transient of the pan/tilt servos and consider them as scaling factors with one step delay. The estimate of $\mathbf{v}$ is defined as

$$\hat{\mathbf{v}}(k) = K_d\mathbf{u}(k-1). \tag{23}$$

Replacing $\mathbf{v}$ and $\mathbf{p}_i$ with $\hat{\mathbf{v}}$ and $\hat{\mathbf{p}}_\mathrm{i}$ in (21), we then can obtain the modified error function as

$$\mathbf{e}(k) = N\left(\hat{\mathbf{p}}_i(k+1), \hat{\mathbf{v}}(k)\right) - N\left(\mathbf{p}_i^*, \hat{\mathbf{v}}(k)\right). \tag{24}$$

The purpose of the design of the tracking control law is to minimize the tracking error function given in (24) by choosing a suitable control input $\mathbf{u}(k)$. Since the dynamics model of the pan/tilt servos is relatively simple, we employ a discrete-time proportional–integral (PI) controller (see, for example, [16]), which is structurally simple but fairly robust. It is very suitable for our real-time application. The incremental implementation of the PI controller is given by

$$\Delta\mathbf{u}(k) = K_p\left[\mathbf{e}(k) - \mathbf{e}(k-1)\right] + \frac{K_pT_s}{T_i}\mathbf{e}(k)$$

where the proportional gain and the integral time are chosen as $K_p = 0.65$ and $T_i = 0.8$, respectively. We note that two identical controllers are respectively used for the pan and tilt servos, since the dynamics of these two servos are very close.

2) Following Control of the UAV: As shown in Fig. 6, to estimate the relative distance between the target and the UAV, we combine the camera model (16) with the transformation in (17) and generate the overall geometric model from an ideal image to the NED frame

$$\mathbf{p}_n = \lambda\mathbf{R}_{n/c}\begin{bmatrix} f_x^{-1} & 0 & 0 \\ 0 & f_y^{-1} & 0 \\ 0 & 0 & 1\end{bmatrix}\begin{pmatrix} x_i \\ y_i \\ 1\end{pmatrix} + \mathbf{t}_{n/c}. \tag{25}$$

We assume that the ground is flat and the height of the UAV to the ground h is known. We have

$$\mathbf{R}_{n/c} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9\end{bmatrix} \qquad \mathbf{t}_{n/c} = \begin{pmatrix} x_{n/c} \\ y_{n/c} \\ z_{n/c}\end{pmatrix} \tag{26}$$

which can be calculated by using the measurements of the onboard navigation sensors. Based on the assumption that the target is on the ground, z_n is equal to zero. We then can derive λ as

$$\lambda = \frac{-z_{n/c}}{r_7\frac{x_i}{f_x} + r_8\frac{y_i}{f_y} + r_9}$$

which, together with (25), yields

$$\begin{pmatrix} x_n \\ y_n \\ z_n\end{pmatrix} = \begin{pmatrix} \lambda\left(r_1\frac{x_i}{f_x} + r_2\frac{y_i}{f_y} + r_3 + x_{n/c}\right) \\ \lambda\left(r_4\frac{x_i}{f_x} + r_5\frac{y_i}{f_y} + r_6 + y_{n/c}\right) \\ 0\end{pmatrix}. \tag{27}$$

As shown in Fig. 6, the relative distance between the target and the UAV is estimated, which is employed as the reference signal to guide the UAV to follow the motion of the target. The tracking reference for the UAV is defined as

$$\begin{pmatrix} x_\mathrm{uav} \\ y_\mathrm{uav} \\ z_\mathrm{uav} \\ \psi_\mathrm{uav}\end{pmatrix}_\mathrm{ref} = \begin{pmatrix} \begin{pmatrix} x_n \\ y_n\end{pmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0\end{bmatrix}\mathbf{R}_{n/b}\begin{pmatrix} c_x \\ c_y \\ 0\end{pmatrix} \\ h_0 \\ \psi_0\end{pmatrix}$$

where c_x and c_y are the desired relative distances between the target and the UAV in the X_b- and Y_b-axes, respectively, h_0 is the predefined height of the UAV above the ground, ψ_0 is the predefined heading angle of the UAV, and $\mathbf{R}_{n/b}$ is the rotation matrix from the body frame to the local NED frame, which can be calculated in terms of the output of the onboard navigation sensors.

VI. Experimental Results

To verify the proposed vision system, multiple tests of the complete system were conducted. During these tests, the proposed vision-based unmanned helicopter SheLion was hovering autonomously at a certain position. If the moving target entered into the view of the onboard camera, the target would be identified and tracked in the video sequence by the vision system automatically. Based on the vision information, the pan/tilt servomechanism was controlled to keep the target in a certain position in the image, as described in Section V-C1. Then, the operator can command the UAV to enter into the following mode, in which the UAV followed the motion of the target autonomously based on the estimated relative distance, using the algorithm proposed in Section V-C2.

The experimental results of the vision-based target detection and tracking in flight are shown in Table I, which indicate that the proposed vision algorithm could effectively identify and track the target in the video sequence in the presence of

TABLE I
EXPERIMENTAL RESULTS OF TARGET DETECTION AND TRACKING IN FLIGHT

Test No.	Total time (s)	Total frames	Target frames detected	Accuracy
1	101.8	761	728	95.66%
2	77.2	591	518	87.65%
3	50.4	388	382	98.45%
4	75.3	572	501	87.59%
5	86.4	662	645	97.43%

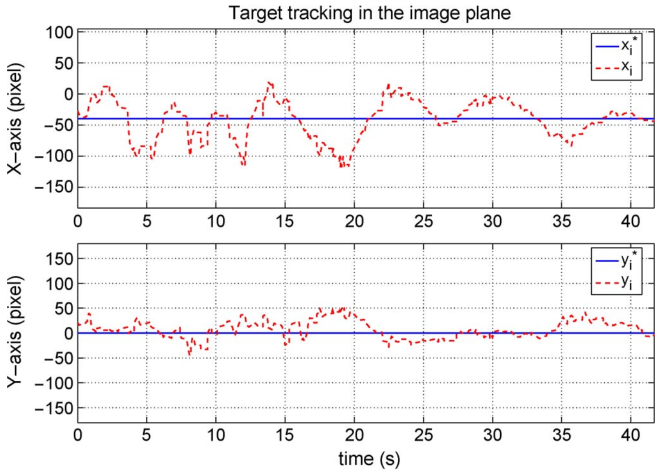


Fig. 7. Test result of the vision-based servo following.

the disturbance of unknown motion between the UAV and the target. One example of the pan/tilt servo tracking control in flight is also shown in Fig. 7. The solid line in Fig. 7 indicates the expected position of the target in the image, and the dashed line indicates the actual location of the target in the image during the flight test. From Fig. 7, we can observe that, in spite of the unknown motion between the UAV and the target, the pan/tilt servomechanism can effectively control target in a boxlike neighborhood of the center point of the image by employing the vision-based pan/tilt servo control.

In the flight tests, the relative distance between the target and the UAV was estimated using the approach presented earlier, which is shown in Fig. 8. The relative distance is also measured using the GPS receiver. The experimental results in Fig. 8 indicate that the vision sensor can provide acceptable relative distance estimates between the UAV and the target based on the altitude information of the UAV and the location of the target in the image.

One example of the ground target following is shown in Fig. 9. In the experiment, the target was manually controlled to move randomly on the flat ground, and the UAV followed the motion of the target automatically based on the scheme proposed in the previous sections. From Fig. 9, we observe that the UAV can follow the trajectory of the target and keep the constant relative distance between the UAV and the target. The results for the moving ground target following of the UAV indicate the efficiency and robustness of the proposed vision-based following scheme. The videos of the vision-based target following tests are available on http://uav.ece.nus.edu.sg/video.html.

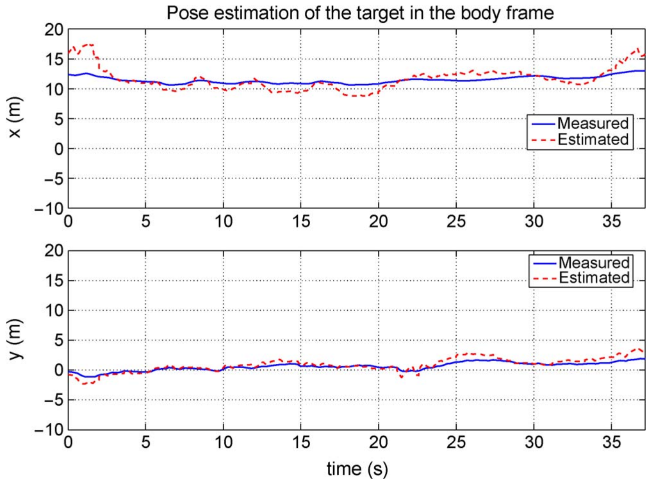


Fig. 8. Test result of the relative distance estimation.

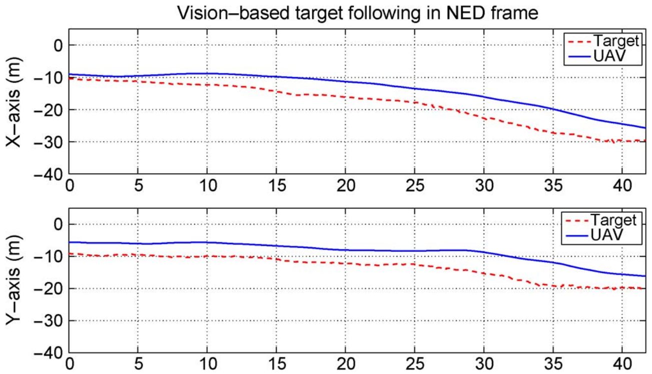


Fig. 9. Test result of the vision-based target following.

VII. CONCLUSION

In this paper, we have presented the comprehensive design and implementation of the vision system for the UAV, including hardware construction, software development, and an advanced ground target seeking and following scheme. Multiple real flight tests were conducted to verify the presented vision system. The experimental results show that this vision system is not only able to automatically detect and track the predefined ground target in the video sequence but also able to guide the UAV to follow the motion of the target in flight. The robustness and efficiency of the developed vision system for UAVs could be achieved by the current system. Our future research focus is to utilize the system for implementing vision-based automatic landing of the UAV on a moving platform in an environment without GPS signals.

REFERENCES

[1] M. J. Adriaans, R. C. Cribbs, B. T. Ingram, and B. P. Gupta, "A160 Hummingbird unmanned air vehicle development program," in *Proc. AHS Int. Specialists' Meeting—Unmanned Rotorcraft: Des., Control Test.*, Chandler, AZ, 2007.

[2] O. Amidi, T. Kanade, and R. Miller, "Vision-based autonomous helicopter research at Carnegie Mellon Robotics Institute 1991–1997," in *Proc. Amer. Helicopter Soc. Int. Conf.*, Gifu, Japan, 1998, pp. 1–12.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, Feb. 1994.

[4] Y. Boykov and D. P. Huttenlocher, "Adaptive Bayesian recognition in tracking rigid objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Hilton Head, SC, 2000, pp. 697–704.

[5] G. R. Bradski and S. Clara, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal Q2 '98*, pp. 1–15, 1998.

[6] G. W. Cai, B. M. Chen, K. M. Peng, M. B. Dong, and T. H. Lee, "Modeling and control of the yaw channel of a UAV helicopter," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3426–3434, Sep. 2008.

[7] R. Canals, A. Roussel, J. L. Famechon, and S. Treuillet, "A biprocessor-oriented vision-based target tracking system," *IEEE Trans. Ind. Electron.*, vol. 49, no. 2, pp. 500–506, Apr. 2002.

[8] M. E. Campbell and W. W. Whitacre, "Cooperative tracking using vision measurements on seascan UAVs," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 613–626, Jul. 2007.

[9] F. Chaumette and S. Hutchinson, "Visual servo control Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[10] F. Chaumette and S. Hutchinson, "Visual servo control Part II: Advanced approaches," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Mar. 2007.

[11] C. Chen and Y. F. Zheng, "Passive and active stereo vision for smooth surface detection of deformed plates," *IEEE Trans. Ind. Electron.*, vol. 42, no. 3, pp. 300–306, Jun. 1995.

[12] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[13] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[14] B. Enderle, "Commercial applications of UAVs in Japanese agriculture," presented at the AIAA 1st Unmanned Aerospace Vehicles, Systems, Technologies, and Operations(UAV) Conf., Portsmouth, VA, 2002, AIAA-2002-3400.

[15] J. Ferruz, V. M. Vega, A. Ollero, and V. Blanco, "Reconfigurable control architecture for distributed systems in the HERO autonomous helicopter," *IEEE Trans. Ind. Electron.*, vol. 58, no. 12, pp. 5311–5318, Dec. 2011.

[16] G. Franklin, J. D. Powell, and A. E. Naeini, *Feedback Control of Dynamic Systems,* 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[17] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 331–340, Apr. 2008.

[18] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 3309–3316.

[19] W. M. Hu, T. N. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[20] Y. Hu, W. Zhao, and L. Wang, "Vision-based target tracking and collision avoidance for two autonomous robotic fish," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1401–1410, May 2009.

[21] M. Isard and A. Blake, "Condensation conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[22] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *J. Aerosp. Comput., Inf. Commun.*, vol. 4, no. 4, pp. 707–738, 2007.

[23] J. Kim and S. Sukkarieh, "SLAM aided GPS/INS navigation in GPS denied and unknown environments," in *Proc. Int. Symp. GNSS/GPS*, Sydney, Australia, 2004.

[24] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking, Part I: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[25] F. Lin, K. Y. Lum, B. M. Chen, and T. H. Lee, "Development of a vision-based ground target detection and tracking system for a small unmanned helicopter," *Sci. China—Series F: Inf. Sci.*, vol. 52, no. 11, pp. 2201–2215, Nov. 2009.

[26] B. Ludington, E. Johnso, and G. Vachtsevanos, "Augmenting UAV autonomy," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 63–71, Sep. 2006.

[27] M. Meingast, C. Geyer, and S. Sastry, "Vision based terrain recovery for landing unmanned aerial vehicles," in *Proc. IEEE Conf. Decision Control*, Atlantis, Bahamas, 2004, pp. 1670–1675.

[28] L. Mejias, S. Saripalli, P. Cervera, and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *J. Field Robot.*, vol. 23, no. 3/4, pp. 185–199, Apr. 2006.

[29] R. Nelson and J. R. Corby, "Machine vision for robotics," *IEEE Trans. Ind. Electron.*, vol. IE-30, no. 3, pp. 282–291, Aug. 1983.

[30] F. Porikli, "Achieving real-time object detection and tracking under extreme conditions," *J. Real-Time Image Process.*, vol. 1, no. 1, pp. 33–40, Oct. 2006.

[31] F. Sadjadi, "Theory of invariant algebra and its use in automatic target recognition," in *Physics of Automatic Target Recognition*, vol. 3. New York: Springer-Verlag, 2007, pp. 23–40.

[32] Z. Sarris and S. Atlas, "Survey of UAV applications in civil markets," in *Proc. 9th Mediterranean Conf. Control Autom.*, Dubrovnik, Croatia, 2001, vol. WA2-A, pp. 1–11.

[33] A. R. Smith, "Color gamut transform pairs," in *Proc. 5th Annu. Conf. Comput. Graph. Interactive Tech.*, New York, 1978, pp. 12–19.

[34] O. Spinka, O. Holub, and Z. Hanzalek, "Low-cost reconfigurable control system for small UAVs," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 880–889, Mar. 2011.

[35] T. Y. Tian, C. Tomasi, and D. J. Heeger, "Comparison of approaches to egomotion computation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, 1996, pp. 315–320.

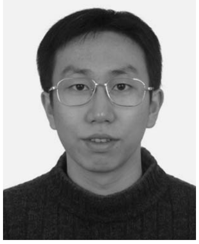
[36] S. Tsugawa, "Vision-based vehicles in Japan: Machine vision systems and driving control systems," *IEEE Trans. Ind. Electron.*, vol. 41, no. 4, pp. 398–405, Aug. 1994.

[37] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos, "Robust target detection and tracking through integration of motion, color and geometry," *Comput. Vis. Image Understand.*, vol. 103, no. 2, pp. 121–138, Aug. 2006.

[38] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

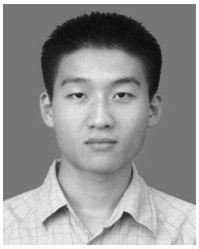
[39] D. Xu, L. Han, M. Tan, and Y. F. Li, "Ceiling-based visual positioning for an indoor mobile robot with monocular vision," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1617–1628, May 2009.

[40] Q. M. Zhou and J. K. Aggarwalb, "Object tracking in an outdoor environment using fusion of features and cameras," *Image Vis.Comput.*, vol. 24, no. 11, pp. 1244–1255, Nov. 2006.



Feng Lin (S'10) received the B.Eng. degree in computer science and control and the M.Eng. degree in system engineering from Beihang University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer and electrical engineering from the National University of Singapore, Singapore, in 2011.

He is currently a Research Associate with Department of Electrical and Computer Engineering, National University of Singapore. His main research interests are unmanned aerial vehicles, vision-based control and navigation, target tracking, robot vision, and embedded vision systems.



Xiangxu Dong received the B.S. degree from the Department of Communications Engineering, Xiamen University, Xiamen, China, in 2006. Since 2007, he has been working toward the Ph.D. degree at the National University of Singapore, Singapore.

His research interests include real-time software, cooperative coordination, formation control, and unmanned aerial vehicles.

Ben M. Chen (S'89–M'92–SM'00–F'07) received the B.S. degree in mathematics and computer science from Xiamen University, Xiamen, China, in 1983, the M.S. degree in electrical engineering from Gonzaga University, Spokane, WA, in 1988, and the Ph.D. degree in electrical and computer engineering from Washington State University, Pullman, in 1991.

From 1983 to 1986, he was a Software Engineer with South-China Computer Corporation, Guangzhou, China, and from 1992 to 1993, he was an Assistant Professor with the Department of Electrical Engineering, The State University of New York, Stony Brook. Since August 1993, he has been with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, where he is currently a Professor. He is the author/coauthor of eight research monographs, including *H_2 Optimal Control* (Prentice Hall, 1995), *H_∞Control and Its Applications* (Springer, 1998; Chinese edition published by Science Press, 2010), *Robust and H_∞ Control* (Springer, 2000), *Linear Systems Theory* (Birkhauser, 2004; Chinese translation published by Tsinghua University Press, 2008), *Hard Disk Drive Servo Systems* (Springer, 2002 and 2006), and *Unmanned Rotorcraft Systems* (Springer, 2011). He served/serves on the editorial boards for a number of international journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica, Systems and Control Letters*, and *Journal of Control Theory and Applications*. His current research interests include robust control, systems theory, unmanned systems, and financial market modeling.

Dr. Chen was the recipient of the Best Poster Paper Award, 2nd Asian Control Conference, Seoul, Korea (in 1997); University Researcher Award, National University of Singapore (in 2000); Prestigious Engineering Achievement Award, Institution of Engineers, Singapore (in 2001); Temasek Young Investigator Award, Defence Science and Technology Agency, Singapore (in 2003); Best Industrial Control Application Prize, 5th Asian Control Conference, Melbourne, Australia (in 2004); Best Application Paper Award, 7th Asian Control Conference, Hong Kong (in 2009); and Best Application Paper Award, 8th World Congress on Intelligent Control and Automation, Jinan, China (in 2010).

Kai-Yew Lum (M'97) received the Diplôme d'Ingénieur from the Ecole Nationale Supérieure d'Ingénieurs Electriciens de Grenoble, Grenoble, France, in 1988 and the M.Sc. and Ph.D. degrees from the Department of Aerospace Engineering, University of Michigan, Ann Arbor, in 1995 and 1997, respectively.

He was with DSO National Laboratories, Singapore, as a Junior Engineer from 1990 to 1993 and then as a Senior Engineer from 1998 to 2001, specializing in control and guidance technologies. In 2001, he joined Temasek Laboratories, National University of Singapore, Singapore, where he is currently the Principal Investigator of research programs in control, guidance, and multiagent systems. His research interests include nonlinear dynamics and control, estimation and optimization, and reduced-order modeling and control in aerodynamics.

Mr. Lum is a member of the IEEE Control Systems Society and the American Institute of Aeronautics and Astronautics.

Tong H. Lee (M'90) received the B.A. degree with first class honors in the Engineering Tripos from Cambridge University, Cambridge, U.K., in 1980 and the Ph.D. degree from Yale University, New Haven, CT, in 1987.

He is currently a Professor with the Department of Electrical and Computer Engineering and the National University of Singapore (NUS) Graduate School for Integrative Sciences and Engineering, National University of Singapore (NUS), Singapore. He was a Past Vice-President (Research) of NUS. He currently holds Associate Editor appointments in *Control Engineering Practice* [an International Federation of Automatic Control (IFAC) journal] and the *International Journal of Systems Science* (Taylor and Francis, London). In addition, he is the Deputy Editor-in-Chief of *IFAC Mechatronics* journal. He has coauthored five research monographs (books) and is the holder of four patents (two of which are in the technology area of adaptive systems, and the other two are in the area of intelligent mechatronics). He has published more than 300 international journal papers. His research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics, and computational intelligence.

Dr. Lee currently holds Associate Editor appointments in the IEEE TRANSACTIONS IN SYSTEMS, MAN, AND CYBERNETICS and IEEE TRANSACTIONS IN INDUSTRIAL ELECTRONICS. He was a recipient of the Cambridge University Charles Baker Prize in Engineering, the 2004 Asian Control Conference (ASCC) (Melbourne) Best Industrial Control Application Paper Prize, the 2009 IEEE International Conference on Mechatronics and Automation Best Paper in Automation Prize, and the 2009 ASCC Best Application Paper Prize. He was an Invited Panelist at the 2000 World Automation Congress, Maui, HI; an Invited Keynote Speaker for the 2003 IEEE International Symposium on Intelligent Control, Houston, TX; an Invited Keynote Speaker for the 2007 International Conference on Life System Modeling and Simulation (LSMS), Shanghai China; an Invited Expert Panelist for the 2009 IEEE International Conference on Advanced Intelligent Mechatronics; an Invited Plenary Speaker for the 2009 International Association of Science and Technology for Development (IASTED) International Conference on Robotics, Telematics and Applications, Beijing, China; an Invited Keynote Speaker for LSMS 2010, Shanghai; and an Invited Keynote Speaker for 2010 IASTED International Conference on Control and Applications, Banff, AB, Canada.