Science in China Series F: Information Sciences

www.scichina.com
info.scichina.com
www.springer.com/scp
www.springerlink.com

Development of a vision-based ground target detection and tracking system for a small unmanned helicopter

LIN Feng, LUM Kai-Yew, CHEN Ben M.† & LEE Tong H.

Department of Electrical & Computer Engineering, National University of Singapore, 117576, Singapore

It is undoubted that the latest trend in the unmanned aerial vehicles (UAVs) community is towards vision-based unmanned small-scale helicopter, utilizing the maneuvering capabilities of the helicopter and the rich information of visual sensors, in order to arrive at a versatile platform for a variety of applications such as navigation, surveillance, tracking, etc. In this paper, we present the development of a vision-based ground target detection and tracking system for a small UAV helicopter. More specifically, we propose a real-time vision algorithm, based on moment invariants and two-stage pattern recognition, to achieve automatic ground target detection. In the proposed algorithm, the key geometry features of the target are extracted to detect and identify the target. Simultaneously, a Kalman filter is used to estimate and predict the position of the target, referred to as dynamic features, based on its motion model. These dynamic features are then combined with geometry features to identify the target in the second-stage of pattern recognition, when geometry features of the target change significantly due to noise and disturbance in the environment. Once the target is identified, an automatic control scheme is utilized to control the pan/tilt visual mechanism mounted on the helicopter such that the identified target is to be tracked at the center of the captured images. Experimental results based on images captured by the small-scale unmanned helicopter, SheLion, in actual flight tests demonstrate the effectiveness and robustness of the overall system.

target tracking, unmanned aerial vehicle, image processing, pattern recognition, Kalman filtering

1 Introduction

Unmanned aerial vehicles (UAVs) have recently aroused much interest in the civil and industrial markets, ranging from industrial surveillance, agriculture, to wildlife conservation[1−4]. Particularly, thanks to its vertical take-off-and-landing, hovering and maneuvering capabilities, the unmanned helicopter has received much attention in the defense and security community[5−9]. More specifically, unmanned helicopters equipped with powerful visual sensors begin to perform a wide range of tasks, such as vision-aided flight control[10,11], tracking[12], terrain mapping[13], and navigation[14,15].

Received February 17, 2009; accepted June 30, 2009
doi: 10.1007/s11432-009-0187-5
†Corresponding author (email: bmchen@nus.edu.sg)
Supported by Temasek Defence Systems Institute of Singapove (Grant No. TDSI/07-003/1A)

Citation: Lin F, Lum K Y, Chen B M, et al. Development of a vision-based ground target detection and tracking system for a small unmanned helicopter. Sci China Ser F-Inf Sci, 2009, 52(11): 2201–2215, doi: 10.1007/s11432-009-0187-5

An important commonality among the various applications of vision on UAVs is vision-based detection and tracking of objects or features. Indeed, many works on the latter have already been reported (see, for example, ref. [16]). Target detection can be considered as a pattern recognition problem. A straightforward approach is to identify the appearance, color and shape signatures of an object captured in image by comparing it with templates in a large library that includes all potential views of the target or its components, in different sizes and orientations[12,17]. However, they involve repeated evaluation of image correlations and are therefore computationally costly. Another tracking approach, called active contour-based tracking, which tracks objects by representing their outlines as contours, and updating these contours dynamically in successive frames via some optimization routine. It is computationally simple, thus allowing efficient implementation in real-time[18]. However, this approach is limited by its sensitivity to the initialization, which make it difficult to automatically detect and track objects[16].

An increasingly popular approach for automatic detection and tracking is the feature-based approach. This involves the segmentation of the object based on its texture, color and appearance, and the extraction of object features by some high-level abstraction, yielding certain descriptors of the object referred to as static features. However, pure static feature detection may fail to detect moving targets as a result of significant variations in the static features caused by nonlinear changes of shapes of the target, noise, distortion, change of lighting condition, and occlusion in the captured images. To overcome the drawback of the static features, an effective combination of detection and motion tracking is needed. Tracking typically involves mathematical tools such as the Kalman filter and Bayesian network (see, for example, refs. [16,19–21]), to provide the behavior or motion of the targets, referred to as dynamic features. While the motion filters provide target-position prediction to aid the detection, the predicted position may cause the target detection algorithm to become trapped into locking onto objects which are close to the predicted position.

Hence, to realize robust target tracking in complex environment, it is necessary to fuse multiple features, including static and dynamic features of a target, under a systematic framework[22]. In this present work, we propose a detection and tracking algorithm in which feature extraction and pattern recognition are based on moment invariants, and a motion model is used to enhance tracking of the maneuvering target. Moment invariant method has the significant advantages of simple calculation and invariance under translation, rotation and scaling of the object in images, caused by the movements of the UAV and a target, first introduced by Hu[12]. However, the original formulation incurs high computation costs. In ref. [24], Chen and Tsai presented improved moment invariants, which were computed along the boundary of a shape and greatly reduced the computation costs.

We present in this work the development of a vision-based ground target detection and tracking system, which is installed onboard a small UAV helicopter. We propose a two-stage pattern recognition technique incorporating an improved moment invariant method to automatically detect ground targets of interest from images captured through the camera mounted on the helicopter. In the proposed approach, the key geometry features of the target are extracted to detect and identify the target, and a Kalman filter is used to estimate and predict the position of the target, referred to as dynamic features. These dynamic features are then combined with geometry features to identify the target in the second-stage of pattern recognition, when geometry features of the target change significantly due to noise and disturbance in the environment. An automatic tracking control scheme is introduced to control the pan/tilt visual mechanism mounted on the helicopter such that the identified target is to be tracked at the center of the captured images. Experimental results based on images captured by the small-scale unmanned helicopter, SheLion, in actual flight tests demonstrate the effectiveness and robustness of the overall system.

The remainder of this paper is organized as fol-

lows: Section 2 describes the configuration of the small-scale unmanned helicopter, SheLion. Section 3 details the vision detection algorithm and motion model. Section 4 describes the target tracking control scheme based on the vision detection. Section 5 shows experiment results in both off-line test and actual flight test. Finally, we draw some concluding remarks in section 6.

2 Brief description of the UAV helicopter platform

The overview of a vision-capable unmanned helicopter, named SheLion, is shown in Figure 1. In this section, we briefly describe the construction of SheLion and its essential components.

The unmanned helicopter, SheLion, consists of three main parts: i) A radio-controlled helicopter, fully equipped for manual operation; ii) an onboard system consisting of navigation and GPS sensors, a wireless modem, a computer for communication, control and data logging, a computer for image processing, and a power supply unit; and iii) a ground station for a user interface and high-level commands, as well as telemetry. Figure 2 shows the overall layout of SheLion. Note that the major components of the onboard system and the ground station of SheLion are similar to a previously reported platform, HeLion[25]. In what follows, we shall focus on the new feature of the vehicle, i.e., the image-processing unit.

The onboard image-processing unit is designed to capture images, to detect objects from the images captured, to estimate the relative position, angle and velocity between the unmanned helicopter and the moving ground target, and finally to send feedback signals to flight control for in-flight tracking of the identified target. In order to achieve these tasks, we construct an image-processing unit with a CMOS camera, a laser pointer, a pan/tilt servo mechanism, a frame grabber and a CPU board for image processing.

Camera and Laser Pointer. To provide a visual evidence of the tracking performance, we assemble a CMOS camera and a laser pointer in parallel and mount them onto a customer-made pan/tilt servo mechanism. The entire assembly is mounted under the fuselage. The pan/tilt servo mechanism consists of two radio control (RC) ser-

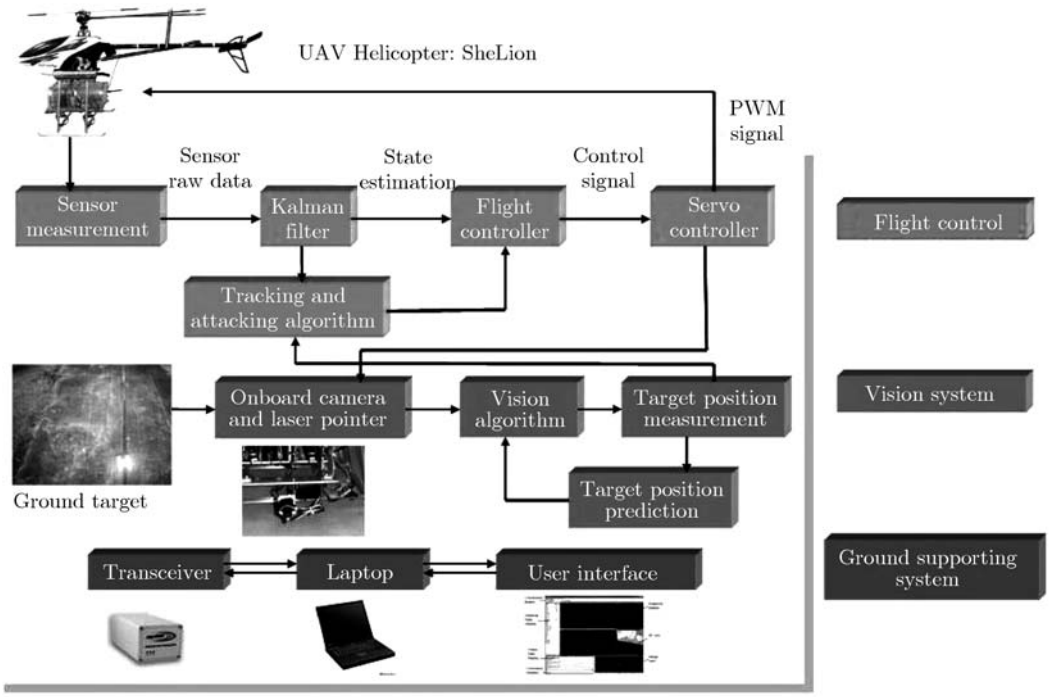

Figure 1 Overview of a vision-capable unmanned helicopter system.

Figure 2 SheLion: a vision-capable unmanned helicopter.

vos, which control the visual sensor for horizontal and vertical rotation respectively. As such, the camera-laser-pointer assembly exhibits two rotational degrees of freedom relative to ground.

Frame Grabber. As we have selected a camera with analog output, a frame grabber is installed onboard to convert the output to digital video signal, and save the converted signal onto the system memory for image processing.

Computer for Image Processing. The computer chosen for this purpose is the Cool Road-Runner III, which is an all-in-one CPU module with specifications similar to those of the PC/104-Plus. The core of the board is a 933 MHz Intel LV Pentium-III processor. We employ such a computer to process the images obtained from the frame grabber, using the image detection technique discussed in the next section. Once a target is identified, this computer will send the necessary information to the flight-control computer, which coordinates the overall mission for in-flight ground-target tracking.

3 Real-time vision detection

As mentioned earlier, a vision detection algorithm based on moment invariants and motion modeling is proposed for SheLion. However, pure moment invariants-based detection may fail to detect moving targets as a result of significant variations in the moment invariants caused by nonlinear changes of shapes of the target, noise, distortion, and occlusion in the captured images. While the motion model provides target-position prediction to aid the detection, the predicted position may cause the vision detection algorithm to become trapped into locking onto objects which are close to the predicted position. In order to check the validity of the motion-model output, a Chi-square test is employed to determine whether the targets are maneuvering. The result of the Chi-square test enables or disables a second-stage pattern recognition that uses the position information and other specified parameters to re-detect the targets. This structure of vision detection is referred to as the two-stage pattern recognition, as shown in Figure 3.

In this section, we describe main modules in the proposed vision detection algorithm: i) image processing; ii) geometry feature extraction; iii) two-stage pattern recognition; iv) motion model; and lastly, v) switching mechanism.

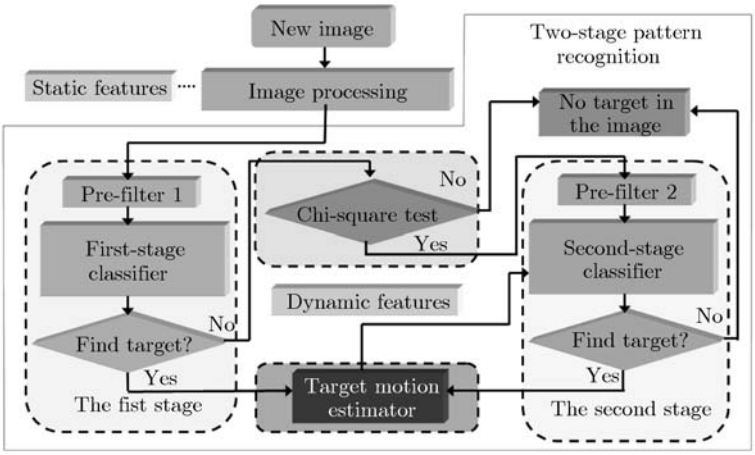

Figure 3 Two-stage pattern recognition structure.

3.1 Image processing

The image-processing module consists of several functions: the median-filter function, the thresholding function and the object segmentation.

Median Filter. A 3×3 median filter is employed in our image-processing algorithm to eliminate, as much as possible, noises in the images such as speckle noise, and salt and pepper noise. Figures 4 and 5 show an original image captured by the onboard frame grabber and that generated by the median filter, respectively.

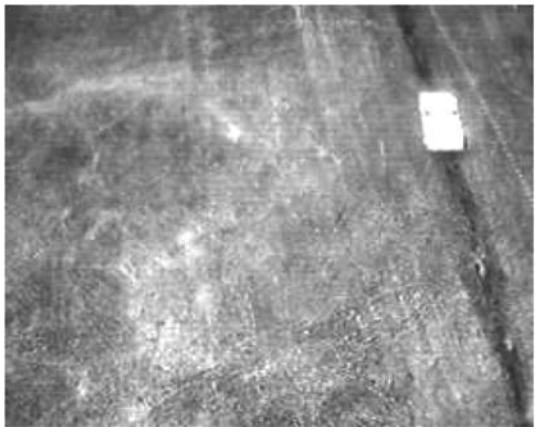

Figure 4 Original image captured by the grabber.

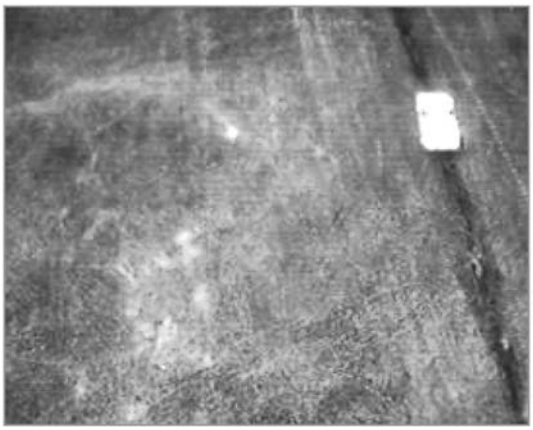

Figure 5 Image generated by the median filter.

Thresholding. The function of the thresholding module is to convert the filtered, gray image into a binary image. Until now we still use a fixed value to threshold all images. An adaptive thresholding algorithm will be developed in the future. For illustration, we show in Figure 6 the output obtained by passing the image in Figure 5 through the thresholding module.

3.2 Object segmentation

After thresholding, several objects may remain in the image. The purpose of object segmentation is to separate all objects in the image from the background. A contour tracking-based algorithm is employed for object segmentation. The purpose of contour tracking is to obtain complete, one pixel-thick boundaries from the black and white image. With this object segmentation, we are able to separate the objects from the background. The output image by object segmentation of Figure 6 is shown in Figure 7.

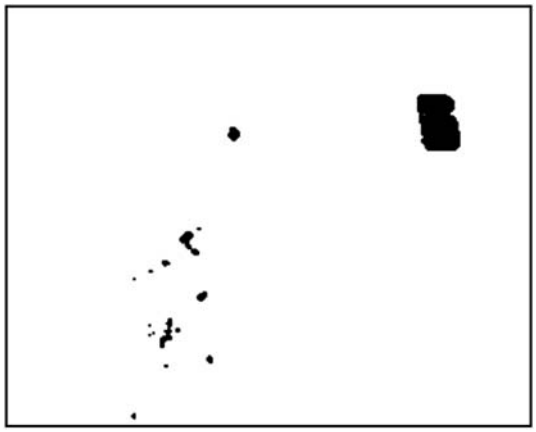

Figure 6 Output image of the thresholding module.

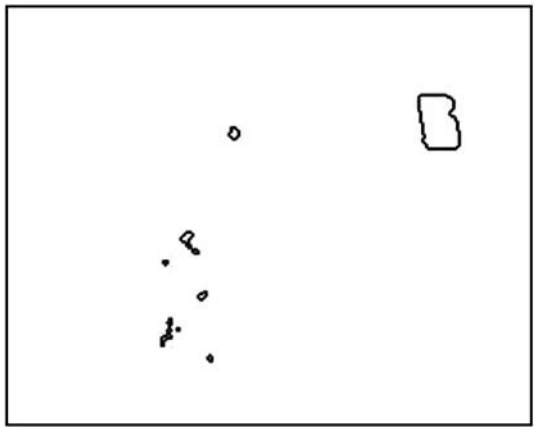

Figure 7 Output image of the object segmentation module.

3.3 Geometry feature extraction

Generally speaking, we are concerned only with the shape of the ground target. In order to have a generalized method capable in detecting targets of arbitrary shapes, we make full use of the geometric information of the target shape, which includes such quantities as perimeter, area, and moments.

Once the geometric information is captured, we then construct a set of descriptors that are invariant under rotation, translation and scaling. In this work, we choose two descriptors: 1) compactness, and 2) moment invariants.

Compactness. Compactness of a shape is measured by the ratio of the square root of the area and the perimeter (in ref. [21] a slightly different definition was given), and is a useful feature for recognition. It can be easily proven that compactness is invariant with respect to translation, scaling and rotation. We use an increment method to calculate area and perimeter in contour tracking, thus reducing computation time.

Moment Invariants. Moment invariants are widely used in image processing, such as pattern recognition, image coding and orientation estimation. In ref. [23], Hu defined the continuous two-dimensional (p, q)-th order moments of a density distribution function $\rho(x, y)$ as

$$m_{pq} = \int_0^\infty \int_0^\infty x^p y^q \rho(x, y) dx dy, \tag{1}$$

where $\rho(x, y)$ is assumed to be a piecewise continuous and bounded function, and has nonzero values only in the finite portion of the XY plane. It can then be proven that the moments of all orders exist and are unique. In particular, we have the following so-called uniqueness theorem[23].

Theorem 3.1. The double moment sequence $\{m_{pq}\}$ is uniquely determined by $\rho(x, y)$; and, conversely, $\rho(x, y)$ is uniquely determined by $\{m_{pq}\}$.

The central moments were defined by Hu as

$$\mu_{pq} = \int_0^\infty \int_0^\infty (x - \bar{x})^p (y - \bar{y})^q \rho(x, y) dx dy, \tag{2}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. It can be easily proven that central moments are invariant to translation. However, they are variant with respect to scaling. In ref. [23], Hu defined normalized central moments, which are invariant with respect to scaling, as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \tag{3}$$

where $\gamma = (p + q)/2 + 1$, $p + q = 2, 3, \ldots$ These normalized central moments are invariant with respect to translation and scaling, but variant with respect to rotation. From the normalized central moments, Hu further constructed a set of moment invariants that are invariant to translation, scaling, and rotation[23]. The four lowest moment invariants are given by

$$\phi_1 = \eta_{20} + \eta_{02}, \tag{4}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \tag{5}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2, \tag{6}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2. \tag{7}$$

While these moment invariants may be used to identify a given shape, their calculation requires every pixel value in the interior and on the boundary of the shape. Thus, the computational cost increases dramatically with an increasing object size.

To reduce the cost of moment computation, in ref. [24], Chen and Tsai presented a new set of moments that only uses the pixels on the boundary of a shape. These moments are given by

$$m_{pq}^c = \int_C x^p y^q \rho(x, y) ds, \quad p, q = 0, 1, 2, \ldots, \tag{8}$$

where C is the boundary curve of the shape, $\int_C$ is a line integral along C, and $ds = \sqrt{(dx)^2 + (dy)^2}$. Since $\rho = 1$ on the boundary and $\rho = 0$ elsewhere, eq. (8) can be simplified as

$$m_{pq}^c = \int_C x^p y^q ds, \quad p, q = 0, 1, 2, \ldots \tag{9}$$

Next, Chen and Tsai defined the new central moments and new normalized central moments as

$$\mu_{pq}^c = \int_C (x - \bar{x})^p (y - \bar{y})^q ds, \quad p, q = 0, 1, \ldots,$$

$$\eta_{pq}^c = \frac{\mu_{pq}^c}{(\mu_{00}^c)^{p+q+1}}, \tag{10}$$

where

$$\bar{x} = \frac{m_{10}^c}{m_{00}^c}, \quad \bar{y} = \frac{m_{01}^c}{m_{00}^c}$$

and

$$\mu_{00}^c = \int_C ds = |C| = \text{length of curve } C.$$

It was shown in ref. [24] that, substituting into eqs. (4)–(7) the new normalized central moments in eq. (10) instead of eq. (3), the resulting moment invariants are also invariant to translation, rotation and scaling.

The formulation of Chen and Tsai can significantly reduce the moment computation cost. However, we have found that the normalization factor μ_{00}^c in eq. (10) is very sensitive to noise. Actually, μ_{00}^c is the length of the boundary curve, compared to which the area of the object is less sensitive to noise, i.e., the variance of the area is smaller than that of the length of the boundary. Hence, we propose a variant of Chen and Tsai's formulation by using the area of the object as the normalization factor. We thus define the improved normalized central moments as

$$\eta_{pq}^m = \frac{\mu_{pq}^c}{A^{(p+q+1)/2}}, \tag{11}$$

where A is the interior area of the object, for $p+q=2,3,\ldots$ In Appendices A and B, we adopt the similar lines of reasoning as in ref. [24] to show that the moment invariants ϕ_i in eqs. (4)–(7), calculated by using η_{pq}^m as normalization factor, are indeed invariant to translation, scaling and rotation.

As an example, we calculate the moment invariant ϕ_1 of a target, captured repeatedly in actual flight, using the normalized central moments in eqs. (10) and (11), respectively. The test results are shown in Figure 8, from which we can see that the distribution of ϕ_1 calculated using η_{pq}^m is narrower than that calculated using η_{pq}^c. The statistical parameters of these two distributions, namely the mean value and standard deviation, are shown in Table 1.

Table 1 Comparison between two normalization methods

	Normalize by μ_{00}	Normalize by area
Mean value	0.01627	1.6763
Standard deviation	0.002119	0.14028
$\frac{\text{Standard deviation}}{\text{Mean value}}$	13.7598%	8.3682%

3.4 Pattern recognition

After feature extraction, we need to choose the target from the objects of interest based on the extracted features: compactness and moment invariants ϕ_i. We use the Bayesian classifier to decide the target. Bayesian classifier is an efficient probabilistic classifier based on Bayes's formula, as described below.

Pre-filter. The purpose of the pre-filter is to improve the robustness of the target recognition and speed up the calculation. This pre-filter removes the objects whose feature values are outside certain regions determined by a priori knowledge.

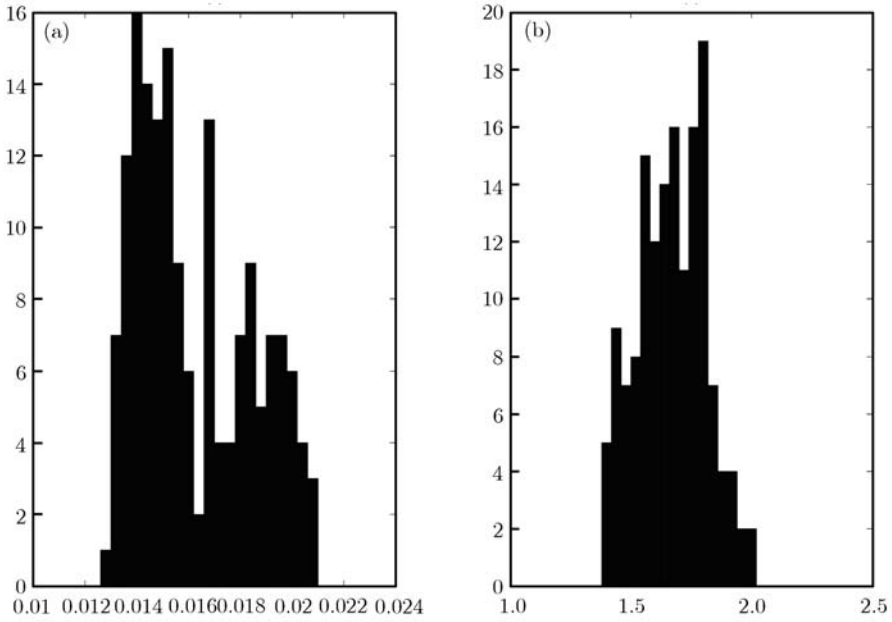

Figure 8 (a) Distribution of ϕ_1 calculated using η_{pq}^c as normalization factor; (b) Distribution of ϕ_1 calculated using η_{pq}^m.

Discriminant Function. The discriminant function is used to estimate the probability of each object being the target, given the feature values of the object and known distributions of the feature values of the target. Hence, the discriminant function is derived from Bayes' theorem as

$$P(\omega_j|\alpha) = \frac{p(\alpha|\omega_j)P(\omega_j)}{p(\alpha)},$$

where ω_j: a discrete class $j, j = 1, 2, \ldots, n$. In our project, we define two classes: a target class and an alternative class; $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)^{\mathrm{T}}$: a continuous feature vector in d-dimensional, which includes a set of continuous feature variables; $P(\omega_j)$: the a priori probability, which is the probability that the target class be present; $P(\omega_j|\alpha)$: the a posteriori probability, which is the probability that α be assigned to class ω_j; $p(\alpha|\omega_j)$: the class-conditional probability density function of the features, given the target class j, which is denoted in low case and represents a function of continuous variable; $p(\alpha)$: a probability density function for the continuous feature variables.

Actually, the purpose of eq. (12) is to decide the a posteriori probability $P(\omega_j|\alpha)$ based on the a priori probability $P(\omega_j)$ and measured feature values of α. Since, given α, the denominator $p(\alpha)$ of eq. (12) does not depend on the classes, it is therefore constant and can be viewed as a scale factor. Thus, we ignore $p(\alpha)$ and only consider the numerator of eq. (12), and define Bayesian discriminant function and the Baysian classifier as

$$f_j(\alpha) = P(\omega_j|\alpha) = p(\alpha|\omega_j)P(\omega_j), \quad (13)$$

$$h(\alpha) = \arg\max_j p(\alpha|\omega_j)P(\omega_j). \quad (14)$$

In our work, we define two classes. The first one is the target class, whereas the second class includes all the objects on the image except the target, called the alternative class. The purpose of the classifier in eq. (14) is to assign the objects to the class corresponding to the largest value of the discriminant function in eq. (13). However, our purpose is, given a set of objects in the image with feature values $(\alpha^1, \alpha^2, \ldots, \alpha^i, \ldots)$, to find the object that is most likely to be the target, assuming that there is a target in the image. This means that we are to choose the object with the largest $f_1(\alpha^i)$. Thus, we modify slightly Bayesian discriminant function and classifier for our application as

$$f_1(\alpha^i) = P(\omega_1|\alpha^i) = p(\alpha^i|\omega_1)P(\omega_1), \quad (15)$$

$$h(\alpha^1, \alpha^2, \ldots, \alpha^i, \ldots) = \arg\max_i p(\alpha^i|\omega_1)P(\omega_1), \quad (16)$$

where α^i is the feature vector of object i. In other words, the function of the classifier is to assign the target class to the object with the largest value of Bayesian discriminant function in eq. (15).

Estimation of the Classifier Parameters. We proceed to estimate the parameters used in Bayesian classifier. The Bayesian classifier in eq. (16) needs to know the distribution of class ω_1: $P(\omega_1)$ and the conditional probability density function of features with the given class ω_1: $p(\alpha|\omega_1)$, where $P(\omega_1)$ is the probability of class ω_1. From the frequency analysis of the training data, we can approximately estimate the value of $P(\omega_1)$. However we can find that in the classifier (16), $P(\omega_1)$ is a fixed value and can be viewed as a factor. So we can set it as a constant value or ignore it.

We use 7 features: *compactness*, ϕ_1, ϕ_2, ϕ_3, ϕ_4, $\tilde{\tilde{x}}$, and $\tilde{\tilde{y}}$, for the classifier to identify the target. The first five features are geometry features of the target referred to as static features. The last two features, $\tilde{\tilde{x}}$, and $\tilde{\tilde{y}}$, are the horizontal and vertical distances between the center of an object and predicted position of the target in the image, which are calculated from images and estimated using the motion model respectively, referred to as dynamic features. The detailed definition of the dynamic feature: $\tilde{\tilde{x}}$ and $\tilde{\tilde{y}}$, will be presented in subsection 3.6. In the first-stage classifier, we employ first five geometry features, and in the second-stage classifier, we employ the total seven features. Furthermore, to simplify the discriminant function, we assume that these features are independent, and the probability densities of *compactness*, $\phi_1, \phi_2, \tilde{\tilde{x}}$, and $\tilde{\tilde{y}}$ are approximately normal distributions, and ϕ_3 and ϕ_4 are Chi-square distributed. We use the maximum likelihood method to obtain the distributions of the features of the target from the training data captured in the flight. The values are shown in Table 2.

Simplified Discriminant Function. Actually, based on the assumption of distribution of

$p(\alpha|\omega_j)$, we take logarithm of the discriminate function in eq. (15), and ignore the constant factors. Thus we can obtain the simplified discriminant function and classifier for the first-stage pattern recognition given by

$$f_1^*(\alpha^i)=\left(\frac{\alpha_1^i}{\sigma_1}\right)^2+\left(\frac{\alpha_2^i}{\sigma_2}\right)^2+\left(\frac{\alpha_3^i}{\sigma_3}\right)^2+\alpha_4^i+\alpha_5^i, \quad (17)$$

$$h^*(\alpha^1,\alpha^2,\dots,\alpha^i,\dots)=\arg\min_i f_1^*(\alpha^i), \quad (18)$$

where $\alpha_1^i,\dots,\alpha_d^i$ are the feature values of object i, and $\sigma_1,\sigma_2,\sigma_3$ are standard deviations of the corresponding features.

Table 2 Distributions of features of the target

feature	distribution
compactness	$N(0.19326,\quad 0.013947^2)$
ϕ_1	$N(1.4267,\quad 0.12582^2)$
ϕ_2	$N(0.1485,\quad 0.13268^2)$
ϕ_3	χ_2^2, (after normalization)
ϕ_4	χ_2^2, (after normalization)
$\tilde{\bar{x}}$	$N(0,\quad 7.9^2)$
$\tilde{\bar{y}}$	$N(0,\quad 7.1^2)$

Discriminant Function Weightings. Indeed, it is unreasonable to assume that all the features are equally important for the target detection. It was observed in the experiment that some features were relatively sensitive to the disturbances, such as distortion of shape, noise in the image and calculation error, while others were not. According to our experience, we thus assign different weightings to the features in the simplified discriminant function, which is defined as

$$f_1'(\alpha^i)=\left(\frac{\alpha_1^i}{\sigma_1}\right)^2 w_1+\left(\frac{\alpha_2^i}{\sigma_2}\right)^2 w_2+\left(\frac{\alpha_3^i}{\sigma_3}\right)^2 w_3 + \alpha_4^i w_4+\alpha_5^i w_5 \quad (19)$$

and

$$h'(\alpha^1,\alpha^2,\dots,\alpha^i,\dots)=\arg\min_i f_1'(\alpha^i), \quad (20)$$

where $w_1,\dots,w_5$ are the weightings of the corresponding features.

3.5 Decision making

After pre-filtering, if there are objects in the image, their discriminant function values are calculated using eq. (19). In terms of these values, a special scheme is defined to identify the target from these objects in the image, as described below.

$$D=\begin{cases}\text{target}=h'(\alpha^1,\alpha^2,\dots,\alpha^i,\dots),\\ \qquad\text{if } \min_i f_1'(\alpha^i)<\Gamma,\\ \text{no target in the image,}\\ \qquad\text{if } \min_i f_1'(\alpha^i)\geqslant\Gamma,\end{cases}$$

where Γ is a thresholding value to be chosen. The scheme above chooses the object, say i, with the smallest value of the simplified discriminant function as the candidate target. If $f_1'(\alpha^i)<\Gamma$, then the scheme decides that object i is the target. Otherwise, the scheme indicates that there is no target in the current image.

3.6 Motion estimation

It is well-known that the motion of a point mass in the two-dimensional plane can be defined by its two-dimensional position and velocity vector. Let $x=[\bar{x},\dot{\bar{x}},\bar{y},\dot{\bar{y}}]^{\mathrm{T}}$ be the state vector of the centroid of the tracked target in the Cartesian coordinate system. Non-maneuvering motion of the target is defined by it having zero acceleration: $[\ddot{\bar{x}},\ddot{\bar{y}}]^{\mathrm{T}}=[0,0]^{\mathrm{T}}$. Strictly speaking, the motion of the intended ground targets may be maneuvering with unknown input. Nevertheless, we assume the standard 4th order non-maneuvering motion model by setting the acceleration as $[\ddot{\bar{x}},\ddot{\bar{y}}]^{\mathrm{T}}=w(t)$, where $w(t)$ is a white noise process[26]. The resulting continuous-time model is

$$\dot{x}=\boldsymbol{A}x+\boldsymbol{B}w,$$
$$z=\boldsymbol{C}x+v,$$

where $v(t)$ is white Gaussian measurement noise, and

$$\boldsymbol{A}=\begin{bmatrix}0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}0&0\\1&0\\0&0\\0&1\end{bmatrix},$$

$$\boldsymbol{C}=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}.$$

Its discrete-time model can be expressed as

$$x_{k+1}=\boldsymbol{\Phi}x_k+\boldsymbol{\Gamma}w_k,$$
$$z_k=\boldsymbol{H}x_k+v_k,$$

where T is the processing time of the vision-based

tracking system, and

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Gamma} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix},$$

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

A Kalman filter can then be designed based on the above motion model to estimate the states of the target in the image plane. The filter consists of the following stages:

1) Predicted state

$$\hat{x}_{k|k-1} = \boldsymbol{\Phi}\hat{x}_{k-1}.$$

2) Updated state estimate

$$\hat{x}_k = \hat{x}_{k|k-1} + \boldsymbol{K}(z_k - \boldsymbol{H}\hat{x}_{k|k-1}).$$

The prediction of the Kalman filter provides important information on the possible location of the target, when the target is partially occluded or changing shape. In terms of the prediction of the Kalman filter, the dynamic feature $\tilde{z}_k$ is defined as

$$\tilde{z}_k = \begin{bmatrix} \tilde{\bar{x}} \\ \tilde{\bar{y}} \end{bmatrix} = z_{ki} - \boldsymbol{H}\hat{x}_{k|k-1},$$

where z_{ki} is the measured locations of the object i in the image; $\boldsymbol{H}\hat{x}_{k|k-1}$ is the predicted location of the target using the Kalman filter.

3.7 Switching mechanism

The purpose of the switching mechanism is to active the second-stage classifier at the moment that the target is lost by the first-stage classifier. The lost of the target can be attribute to the poor match of features in the first-stage classifier due to noise, distortion, or occlusion in the image. In addition, an alternative reason may be the maneuvering motion of the target, which makes it out of the image. Therefore, in order to know the reason and take the special way to find 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.

We consider the estimation error as a random variable, which is defined by

$$\begin{aligned} \varepsilon &= ||\boldsymbol{H}\hat{x}_{k-1} - z_{k-1}||^2_{\Sigma^{-1}} \\ &= (\boldsymbol{H}\hat{x}_{k-1} - z_{k-1})'\Sigma^{-1}(\boldsymbol{H}\hat{x}_{k-1} - z_{k-1}), \end{aligned}$$

where $\boldsymbol{H}\hat{x}_{k-1} - z_{k-1}$ is assumed to be $N(0, \Sigma)$-distributed. ε is Chi-square distributed with 2 degrees of freedom (x and y directions) under H_0.

$$\begin{cases} \varepsilon < \lambda = \chi^2_2(\alpha), & H_0 \text{ is true}, \\ \varepsilon \geqslant \lambda = \chi^2_2(\alpha), & H_1 \text{ is true}, \end{cases}$$

where $1-\alpha$ is the level of confidence, which should be sufficient high (for our project, $1-\alpha = 99\%$). If H_0 is true, the Chi-square testing-based switching declares the target is still in the image and enables the second-stage pattern recognition.

3.8 Second-stage pattern recognition

In fact, the structure of the second-stage classifier is similar to the first-stage classifier, albeit with different parameters:

1) increased regions of the pre-filter to allow for possible shape distortion of the target;

2) increased threshold value for the discriminant function;

3) the dynamic features are used in the discriminant function, which is defined as below:

$$\begin{aligned} f'_1(\alpha_i) = (\alpha^i_1)^2 w_1 + (\alpha^i_2)^2 w_2 + (\alpha^i_3)^2 w_3 + \alpha^i_4 w_4 \\ + \alpha^i_5 w_5 + (\alpha^i_6)^2 w_6 + (\alpha^i_7)^2 w_7 \end{aligned} \tag{21}$$

and

$$h'(\alpha^1, \alpha^2, \ldots, \alpha^i, \ldots) = \arg\min_i f'_1(\alpha^i), \tag{22}$$

where $\alpha^i_6 = \tilde{\bar{x}}$, and $\alpha^i_7 = \tilde{\bar{y}}$, which define the vertical and horizontal distances between the center of object i and the predicted location of the target in images respectively;

4) a different weighting matrix, which gives higher weightings to the dynamic features.

The purpose of the second stage classifier is to combine dynamic features with geometry features of the target to detect it under noise and occlusion condition. Since the geometry features may change significantly under such condition, it is meaningful to give higher weightings to the dynamic features.

4 Target tracking

After detecting the target in the image, the visual tracking control system is proposed to control the pan/tilt servo mechanism to minimize a tracking error function, which is also called eye-in-hand visual servoing[27,28]. Since the relative distance between the UAV and the ground target is large, if the error function is defined in any 3D reference coordinate frame, coarse estimation of the relative pose between the UAV and the ground target may cause the moving target falls out of the visual field the visual sensor, while adjusting the pan/tilt servo mechanism, and also affect the accuracy of the pose reached after convergence. In our project, to make tracking control more robust and stable, we define a tracking error function in the visual sensor frame called image-based visual servo control, which is given by

$$e(t) = P_m - P_m^* = \begin{pmatrix} x_m \\ y_m \end{pmatrix} - \begin{pmatrix} x_m^* \\ y_m^* \end{pmatrix}, \qquad (23)$$

where P_m and P_m^* are the measured and desired locations of the centroid of the tracked target with respect to the image plane respectively. In our project, we set $P_m^* = [0,0]^{\mathrm{T}}$, which is the center point of the captured image.

4.1 Tracking control scheme

Once the error function is selected, a robust closed-loop tracking control scheme can be designed as illustrated in Figure 9. In Figure 9, L is the kinematic relationship between the output of the pan/tilt servo mechanism V and variation of the coordinates of the target P_m in the image plane; $W = [\phi_w, \theta_w]^{\mathrm{T}}$ is the output of the kinematic inverse of L; $U = [\phi_u, \theta_u]^{\mathrm{T}}$ is the output of the tracking controller; $V = [\phi_v, \theta_v]^{\mathrm{T}}$ is the output of the pan/tilt servo mechanism. As shown in Figure 9, to achieve image-based visual servo control, we must know the relationship between the variation of P_m and output of the pan/tilt servo mechanism as well as the dynamic of the pan/tilt servo mechanism.

4.2 Models of the pan/tilt servos

It can be shown that

$$P_m = L(V, P_{m0}) = MP_c/(e_3^{\mathrm{T}} P_c), \qquad (24)$$

where P_{m0} is an equivalent point in the image plane; $e_3 = [0,\ 0,\ 1]^{\mathrm{T}}$; M is the camera calibration matrix. In term of the pinhole model for the perspective projection of the visual sensor, without loss of generality, we define $M = \mathrm{diag}\{f_x, f_y\}$; P_c is the coordinate of the centroid of the target with respect to the camera coordinate system, which is given by

$$P_c = R_y R_x P_{c0}, \qquad (25)$$

$$R_x = R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_v & -\sin\phi_v \\ 0 & \sin\phi_v & \cos\phi_v \end{bmatrix}, \qquad (26)$$

$$R_y = R_\theta = \begin{bmatrix} \cos\theta_v & 0 & \sin\theta_v \\ 0 & 1 & 0 \\ -\sin\theta_v & 0 & \cos\theta_v \end{bmatrix}, \qquad (27)$$

where P_{c0} is the coordinate of an equivalent point in the camera coordinate system, which is corresponding to the equivalent point P_{m0} in the image plane. In this paper, P_{m0} is chosen as $[0,\ 0]^{\mathrm{T}}$, which is the center of the image, and $P_{c0} = [0,\ 0,\ z_{c0}]^{\mathrm{T}}$ is a point on the z-axis of the camera coordinate.

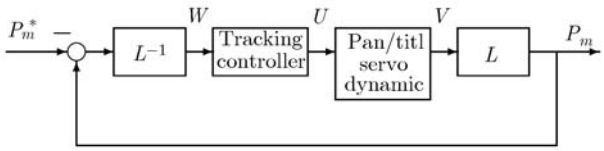

Figure 9 Block diagram of the tracking control scheme.

Substituting eqs. (26) and (27) into eq. (25), we can obtain

$$P_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} \sin\theta_v \cos\phi_v \\ -\sin\phi_v \\ \cos\theta_v \cos\phi_v \end{pmatrix} z_{c0}. \qquad (28)$$

Thus, using eqs. (28) and (24), we can immediately obtain the relationship between the variant of the location of the target in the image plane and the output of the pan/tilt servo mechanism:

$$P_m = \begin{pmatrix} x_m \\ y_m \end{pmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{pmatrix} x_c \\ y_c \end{pmatrix} \frac{1}{z_c} = \begin{pmatrix} \dfrac{f_x \sin\theta_v}{\cos\theta_v} \\ \dfrac{-f_y \sin\phi_v}{\cos\theta_v \cos\phi_v} \end{pmatrix}. \qquad (29)$$

The pan/tilt servos can be approximately considered as two decoupled servos controlling the visual sensor for horizontal and vertical rotation respectively. The dynamic model of each of them consists of a simply feedback loop and a saturation part of an amplifier, which is shown in Figure 10. To identify the parameters of the model, we inject step signals with small and big values to the pan/tilt servos to estimate the parameters of the linear part and saturation part respectively. The parameters of the models of the vertical and horizontal servos are given in Table 3.

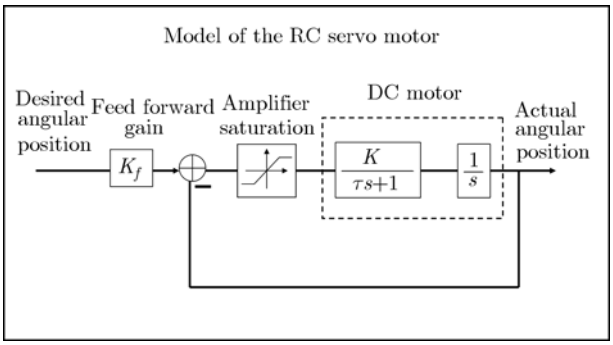

Figure 10 The nonlinear model of the RC servo model.

Table 3 Estimated parameters through the system identification approach

Parameters	Tilt servo	Pan servo
K_f	1.2052	0.98256
τ	0.0229	0.2277
K	38.3406	119.9939
Saturation	6.25	15.0
Time delay (ms)	120	120

Finally, we note that the kinematic inverse of L is given by

$$W = \begin{pmatrix} \phi_w \\ \theta_w \end{pmatrix} = L^{-1}(e) = \begin{pmatrix} \tan^{-1}\dfrac{x_m - x_m^*}{f_x} \\ \tan^{-1}\dfrac{-(y_m - y_m^*)f_x}{f_y\sqrt{f_x^2 + (x_m - x_m^*)^2}} \end{pmatrix}.$$

4.3 PI control

The purpose of the design of the tracking control law is to minimize the error function given in eq. (23) by choosing a suitable control input $u(k)$. Since the dynamics model of the pan/tilt servos is relatively simple, we employ a discrete-time PI controller (see, for example, ref. [29]), which is structurally simple but fairly robust. It is very suitable for our real-time application. The PI controller is given by

$$u(k) = k_p L^{-1}(e(k)) + k_i T_s \sum_{i=1}^{k} L^{-1}(e(i)),$$

where the proportional and integral gains are chosen as $k_p = 1$ and $k_i = 0.75$, respectively. We note that two identical controllers are respectively used for the pan and tilt servos.

5 Experimental results

In an off-line experiment using flight-test data, we tested the vision detection algorithm in the detection of a radio-controlled toy car. Two sets of data were available: the training data and the testing data, both of which were collected in flight. The results are summarized below and in Table 4:

• Training: The total number of training data was 539 frames. Without the second-stage classifier, the algorithm failed to detect the toy car in 88 frames out of the 539 frames. In contrast, with the second-stage classifier, the algorithm failed to detect the toy car in 27 frames out of the 539 frames. The accuracy was therefore increased by 12.33% during training.

• Testing: The total number of testing data was 426 frames, without the second-stage classifier, the algorithm failed to detect the toy car in 153 frames out of the 426 frames. In contrast, with the second-stage classifier, the algorithm failed to detect the toy car in 84 frames out of the 426 frames. The accuracy was therefore increased by 16.2% during testing.

The experimental results indicated that aided by the second-stage classifier, the vision algorithm could effectively re-detect the target, which was lost by the first-stage classifier. This can be adequately explained by the switching mechanism, which intelligently chooses the suitable classifier under different situation, and which makes sure of detection of the correct target in the initial frames and tracking of the target under disturbance.

The proposed vision-based tracking algorithm

Table 4 Experimental results using training and test data

	No of images	No of errors	Accuracy
Training data without 2nd pattern recognition	539	88	83.67%
Training data with 2nd pattern recognition	539	27	95.00%
Testing data without 2nd pattern recognition	426	153	64.08%
Testing data with 2nd pattern recognition	426	84	80.28%

Table 5 Experimental results of the visual tracking tests

Times	Total frame	Detected frames	Accuracy
1	219	191	87.21%
2	284	209	73.59%
3	703	538	76.53%
4	375	295	78.67%
5	676	508	75.15%
6	431	311	72.16%
7	108	91	84.26%
8	1544	1162	75.26%
9	646	529	81.89%

is implemented in the onboard system of the unmanned helicopter, SheLion. The processing rate of the algorithm is 16 fps. During the real flight tests, the helicopter was manually controlled to hover at a fixed position 10 meters above the flat ground, and the onboard visual tracking system automatically identified and tracked the ground moving target: a toy car, which was manually controlled to randomly move in the flat ground.

We performed 9 times of visual tracking tests and the tracking results are shown in Table 5. During these tests, the visual tracking system can successfully track the ground target. One example of the tracking errors in vertical and horizontal direction is shown in Figure 11, which indicates that the tracking error is bounded. The experimental results demonstrate the robustness and effectiveness of the visual tracking system, which can automatically identify and track the moving target in the real flight.

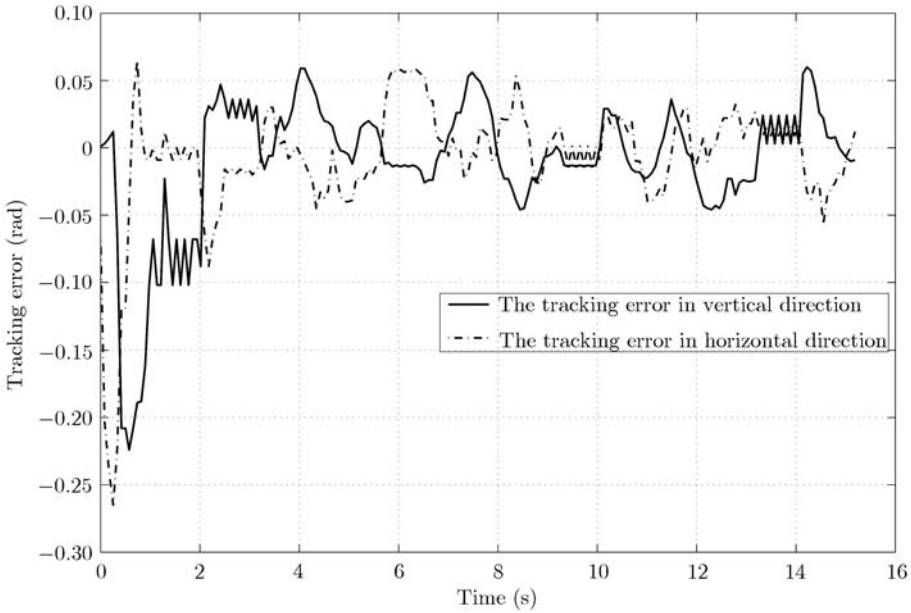

Figure 11 The tracking error of θ_c and ϕ_c.

6 Conclusion

We have presented in a vision-based ground target detection and tracking system for a small UAV helicopter. The experimental results show that the system is capable of automatically detecting the ground target of interest, and effectively track the detected target at the center of the camera screen in spite of the movement of the ground target and the UAV platform. However, there is still room for improvement. We are currently focusing on integrating the system with helicopter navigation and control as well as ego-motion estimation, in order to realize autonomous in-flight ground target detection, tracking and attacking.

1 Sarris Z, Atlas S. Survey of UAV applicatins in civil markets. In: Proceedings of the 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia, 2001. 1–11

2 Herwitz S R, Dunagan S, Sullivan D, et al. Solar-powered UAV mission for agricultural decision support. In: Proceedings of Geoscience and Remote Sensing Symposium, Toulouse, France, 2003. 1692–1694

3 Ludington B, Johnson E, Vachtsevanos G. Augmenting UAV autonomy. IEEE Rob Autom Mag, 2006, 13(3): 63–71

4 Campbell M E, Whitacre W W. Cooperative tracking using vision measurements on seascan UAVs. IEEE Trans Control Syst Technol, 2007, 15(4): 613–626

5 Santana P, Barata J. Unmanned helicopters applied to humanitarian demining. In: Proceedings of 10th IEEE Conference on Emerging Technologies and Factory Automation, Catania, Italy, 2005. 729–738

6 Cai G W, Chen B M, Peng K M, et al. Modeling and control system design for a UAV helicopter. In: Proceedings of the 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 2006. 1–6

7 Gavrilets V, Shterenberg A, Dahleh M A, et al. Avionics system for a small unmanned helicopter performing aggressive maneuvers. In: Proceedings of the 19th Digital Avionics Systems Conferences, Philadelphia, USA, 2000. 1–7

8 Roberts J M, Corke P, Buskey G. Low-cost flight control system for a small autonomous helicopter. In: Proceedings of the 2002 Australian Conference on Robotics and Automation, Auckland, New Zealand, 2002. 546–551

9 Sprague K, Gavrilets V, Dugail D, et al. Design and applications of an avionic system for a miniature acrobatic helicopter. In: Proceedings of the 20th Digital Avionics Systems Conferences, Daytona Beach, USA, 2001. 1–10

10 Guenard N, Hamel T, Mahony R A. Practical visual servo control for an unmanned aerial vehicle. IEEE Trans Rob, 2008, 24(2): 331–340

11 Amidi O, Kanade T, Miller R. Vision-based autonomous helicopter research at carnegie mellon robotics institute 1991–1997. In: Proceedings of American Helicopter Society International Conference, Gifu, Japan, 1998. 1–12

12 Mejias L, Saripalli S, Cervera P, et al. Visual servoing of an autonomous helicopter in urban areas using feature tracking. J Field Rob, 2006, 23(3): 185–199

13 Meingast M, Geyer C, Sastry S. Vision based terrain recovery for landing unmanned aerial vehicles. In: Proceedings of IEEE Conference on Decision and Control, Atlantis, Bahamas, 2004. 1670–1675

14 Hrabar S, Sukhatme G S, Corke P, et al. Combined optic-flow and stereobased navigation of urban canyons for a UAV. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. 3309–3316

15 Kim J, Sukkarieh S. Slam aided gps/ins navigation in gps denied and unknown environments. In: The 2004 International Symposium on GNSS/GPS, Sydney, Australia, 2004. 1–14

16 Hu W M, Tan T N, Wang L, et al. A survey on visual surveillance of object motion and behaviors. IEEE Trans Syst Man Cybern, 2004, 34(3): 334–352

17 Sadjadi F. Theory of invariant algebra and its use in automatic target recognition. Phys Autom Target Recognit, 2007, 3: 23–40

18 Sattigeri R, Johnson E, Calise A, et al. Vision-based target tracking with adaptive target state estimator. In: Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, USA, 2007. 1–13

19 Johnson E N, Calise A J, Watanabe Y, et al. Real-time vision-based relative aircraft navigation. J Aerosp Comput Inf Commun, 2007, 4(4): 707–738

20 Betser A, Vela P, Tannenbaum A. Automatic tracking of flying vehicles using geodesic snakes and kalman filtering. In: Proceedings of 43rd IEEE Conference on Decision and Control, Atlantis, Bahama, 2004. 1649–1654

21 Zhou Q M, Aggarwalb J K. Object tracking in an outdoor environment using fusion of features and cameras. Image Vision Comput, 2006, 24(11): 1244–1255

22 Veeraraghavan H, Schrater P, Papanikolopoulos N. Robust target detection and tracking through integration of motion, color and geometry. Comput Vis Image Und, 2006, 103(2): 121–138

23 Hu M K. Visual pattern recognition by moment invariants. IEEE Trans Inf Theory, 1962, 8(2): 179–187

24 Chen C C, Tsai T I. Improved moment invariants for shape discrimination. Pattern Recogn, 1993, 26(5): 683–686

25 Cai G W, Peng K M, Chen B M, et al. Design and assembling of a UAV helicopter system. In: Proceedings of International Conference on Control and Automation, Budapest, Hungary, 2005. 697–702

26 Li X R, Jilkov V P. Survey of maneuvering target tracking, Part I: Dynamic models. IEEE Trans Aerosp Electron Syst, 2003, 39(4): 1333–1364

27 Chaumette F, Hutchinson S. Visual servo control part I: Basic approaches. IEEE Rob Autom Mag, 2006, 13(4): 82–90

28 Chaumette F, Hutchinson S. Visual servo control part II: Advanced approaches. IEEE Rob Autom Mag, 2007, 14(1): 109–118

29 Franklin G, Powell J D, Emami-Naeini A E. Feedback Control of Dynamic Systems. 4th ed. New Jersey: Prentice-Hall, 2002

Appendix A Proof of moment invariants remain unchanged under scaling

Theorem A.1. Suppose a shape has smooth boundary C, and this shape has been homogeneously rescaled by a factor r, then we get new boundary C'. Then

$$(\eta_{pq}^m)' = \eta_{pq}^m, \tag{A1}$$

where

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

and where A is the area of the original object, and A' is the area of the rescaled object.

Proof. The lines of reasoning in the following proof follow from ref. [24]. For any $r > 0$, we can get the relationship between $(\mu_{pq}^c)'$ and μ_{pq}^c, which is given by

$$\begin{aligned}(\mu_{pq}^c)' &= \int_{C'} [x(s')]^p [y(s')]^q ds' \\ &= \int_{C'} [rx(s)]^p [ry(s)]^q d(rs) \\ &= r^{p+q+1} \mu_{pq}^c.\end{aligned}$$

Note that the relationship between area of the original object and area of the rescaled object is given by

$$A' = r^2 A. \tag{A2}$$

Thus, for any $r > 0$, we have

$$\begin{aligned}(\eta_{pq}^m)' &= \frac{(\mu_{pq}^c)'}{(A')^{(p+q+1)/2}} = \frac{r^{p+q+1}\mu_{pq}^c}{(r^2 A)^{(p+q+1)/2}} \\ &= \frac{\mu_{pq}^c}{A^{(p+q+1)/2}} = \eta_{pq}^m,\end{aligned}$$

which shows that η_{pq}^c are indeed invariant with respect to homogeneous scaling.

Appendix B Proof of moment invariants remain unchanged under rotation

Theorem B.1. Suppose that C is a smooth boundary curve of the original object and C' is the boundary curve obtained by rotating the object an angle θ. Then

$$\phi_i' = \phi_i, \qquad 1 \leqslant i \leqslant 4,$$

where ϕ_i and ϕ_i' are defined in eqs. (4) to (7) calculated by using η_{pq}^m and $(\eta_{pq}^m)'$ for $p + q = 2, 3, \ldots$

Proof. Note that

$$\begin{aligned}(\mu_{pq}^c)' &= \int_{C'} [x(s')]^p [y(s')]^q ds' \\ &= \int_C [x(s)\cos\theta - y(s)\sin\theta]^p \\ &\quad \times [y(s)\sin\theta + x(s)\cos\theta]^q ds.\end{aligned}$$

Since $ds' = ds$, and the area of the object does not change under the rotation, i.e., $A' = A$, from eq. (4), we have

$$\begin{aligned}(A')^{1.5}\phi_1' &= (A')^{1.5}((\eta_{20}^m)' + (\eta_{02}^m)') \\ &= (\mu_{20}^c)' + (\mu_{02}^c)' \\ &= \int_C \{[x(s)\cos\theta - y(s)\sin\theta]^2 \\ &\quad + [x(s)\sin\theta + y(s)\cos\theta]^2\} ds \\ &= \int_C \{[\cos^2\theta + \sin^2\theta][x(s)]^2 \\ &\quad + [\cos^2\theta + \sin^2\theta][y(s)]^2\} ds \\ &= \int_C [x(s)]^2 + [y(s)]^2 ds \\ &= \mu_{20}^c + \mu_{02}^c = (A)^{1.5}(\eta_{20}^m + \eta_{02}^m) \\ &= (A')^{1.5}\phi_1.\end{aligned}$$

Therefore

$$\phi_1' = \phi_1.$$

Similarly, the same can be proven for $\phi_i' = \phi_i$, for $i = 2, \ldots, 4$.