Robust Vision-Based Target Tracking Control System for an Unmanned Helicopter Using Feature Fusion

Feng Lin, Ben M. Chen, Tong H. Lee
Department of Electrical & Computer Engineering, National University of Singapore
4 Engineering Drive 3, Singapore 117576
{linfeng, bmchen, eleleeth}@nus.edu.sg

Abstract

We present in this paper a robust vision-based tracking control system for an unmanned helicopter to track a moving ground target. It integrates a real-time vision-based target detection algorithm with a tracking control law in a closed loop. First, the proposed target detection algorithm extracts geometry, color and motion features from captured images. Based on these features, a finite-state machine is then introduced to dynamically coordinates the work of decision making under the Bayes framework, and a tracking control law is designed to minimize a certain tracking error function. Experimental results obtained from actual flight tests are also presented and demonstrate the effectiveness and robustness of our vision-based tracking control system in real scenes.

1 Introduction

In the last decade, unmanned helicopters equipped with powerful visual sensors begin to perform a wide range of tasks, such as vision-aided flight control [1], tracking [2], terrain mapping [3], and navigation [4]. A basic task among the various applications of vision on UAVs is vision-based detection and tracking of objects or features [5]. The features of objects, straightforwardly, can be extracted from photo-metrical appearance of the objects, such as image templates, geometry, color, texture and many more, which are 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, the behavior or motion of the targets, referred to as the dynamic feature, is also taken into account in the applications of the moving target tracking. The dynamic feature can typically be derived from mathematical tools such as the Kalman filter, Bayesian network, particle filter (see, for example, [6, 7]). 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, under a systematic framework [8]. Typically, these features are fused under from the Bayes framework to neural network to do the pattern recognition. In addition to the pure tracking in the image sequences, it is more attractive to integrate vision information with the control strategy in the closed loop, which is referred to as vision-based servoing.

The main contribution of this paper is to present a novel framework to realize robust vision-based ground target detection for a UAV, and combine the vision target detection with the control strategy to achieve vision-based tracking in real scenes. The proposed target detection algorithm fuses geometry, color and motion features which are described by moment invariants, a color histogram, and motion estimation of a Kalman filter respectively. A finite-state machine coordinates the work of decision making under the Bayes framework by choosing features and adjust the weightings of different features in the discriminant function dynamically, which is also a contribution of this paper. Second, a tracking error function is proposed and a proportional-integral tracking control law is designed and implement in the on-board system to achieve robust tracking control.

The remainder of this paper is organized as follows. Section 2 and 3 detail the vision-based target detection algorithm and the design of the tracking control law. Section 4 describes experiment results of vision-based tracking control in actual flight tests. The last section summarizes the paper and discusses the future work.

2 The target detection algorithm

The overall structure of the proposed target detection algorithm is illustrated in Figure 1, which consists of two main parts: image processing and decision making.

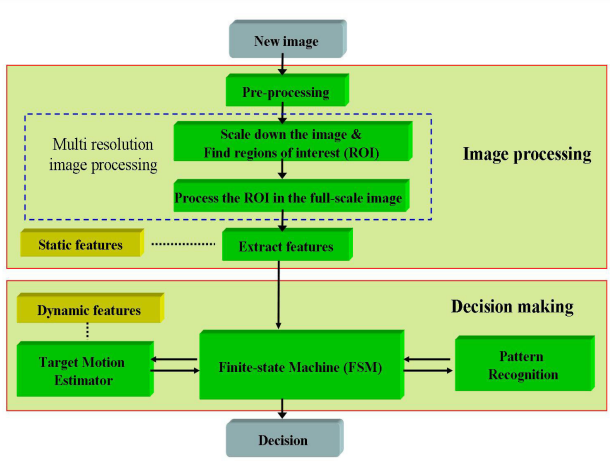

Figure 1: The Structure of the target detection algorithm

2.1 Image processing

The purpose of the image processing is to separate the foreground objects from the background in the images and extract their static features for the pattern recognition.

2.1.1 Pre-processing

A threshold segmentation approach is applied to the captured color image based the assumptions that the target has a distinct color distribution compared to the surrounding background. To make the surface color of the target constant and stable under the varying lighting condition, the color image is represented in the HSV space[9], which stands for Hue (hue), Saturation (sat) and Value (val). We apply pre-defined threshold ranges to hue, sat, and val channels: hue_r = [h_1, h_2], sat_r = [s_1,s_2], val_r = [v_1, v_2]. Only the pixel values falling in these color ranges are described as foreground points.

2.1.2 Multi-resolution image processing

The multi-resolution image processing aims to remove noise and false objects, as well as obtains smooth and complete contour of the objects falling in the regions of interest in the segmented image. This processing uses morphological operations and contour tracking, and processes images from coarse to fine to save computation time.

2.1.3 Geometry feature extraction

The four lowest moment invariants are employed to describe the geometry features of the objects, which are independent of position, size and orientation in the visual field. The four lowest moment invariants is defined based on the boundary curve of the shape C in the segmented image $I(x, y)$, which is given by

$$\begin{aligned}\phi_1 &= \eta^m_{20} + \eta^m_{02} \\ \phi_2 &= (\eta^m_{20} - \eta^m_{02})^2 + 4(\eta^m_{11})^2 \\ \phi_3 &= (\eta^m_{30} - 3\eta^m_{12})^2 + (\eta^m_{03} - 3\eta^m_{21})^2 \\ \phi_4 &= (\eta^m_{30} + \eta^m_{12})^2 + (\eta^m_{03} + \eta^m_{21})^2\end{aligned}$$

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

$$\eta^m_{pq} = \frac{\mu^c_{pq}}{A^{(p+q+1)/2}}$$

where A is the interior area of the shape; μ^m_{pq} are the central moments defined as below:

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

where $ds = \sqrt{(dx)^2 + (dy)^2}$, and $[\bar{x},\ \bar{y}]$ is the coordinate of the centroid of the shape in the image plane.

2.1.4 Color feature extraction

We also employ color histogram to represent the color distribution 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 be implemented in the real-time image processing system. In the proposed color histogram, hue and val are employed to constructed the color histogram for object recognition, which is formally defined by:

$$\begin{aligned}H &= \{h(i,j)\}_{i=1,\ldots,N_h;j=1,\ldots,N_v} \\ h(i,j) &= \sum_{(x,\ y)\in\Omega} \delta(i, [\frac{hue_{(x,y)}}{N_h}])\delta(j, [\frac{val_{(x,y)}}{N_v}]))\end{aligned}$$

where Ω is the region of the target, N_h, N_v are the partition numbers of hue and val color channels, and $\delta(a, b)$ is the Kronecker delta function defined by:

$$\delta(a,b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{elsewhere} \end{cases}$$

In order to classify the target and other false objects based on the color distribution, the color histogram intersection is employed [10] to match the color histogram of each object with the pre-defined target template. The color histogram intersection is defined by:

$$d(H,G) = \frac{\sum_{i=1}^{N_h}\sum_{j=1}^{N_v} \min(H(i,j), G(i,j))}{\min(|H|, |G|)}$$

where $|H|$ and $|G|$ are the numbers of the pixels in the image region H and G. The advantage of this distance formula is that the colors not present in the defined target histogram do not contribute to the intersection distance. Then the effect of the background to the intersection distance can be reduced.

2.2 Decision making

After we extract above static features from the foreground objects, we also calculate the dynamic motion using a Kalman filter based on the target's motion model. Both of the static and dynamic features extracted are employed in the pattern recognition.

2.2.1 Motion model

The motion of the centroid of the target: $x = [\bar{x}, \dot{\bar{x}}, \bar{y}, \dot{\bar{y}}]^{\mathrm{T}}$ in the two-dimensional image coordinate is tracked using a standard 4th-order Kalman filter, which predicts the possible location of the target in the successive frames. Since the visual sensor is attached to the moving pan/tilt servos, the predicted location of the centroid of the target is compensated by the motion of the servos, which is defined by:

$$z_k = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}_k + \begin{bmatrix} f_c & 0 \\ 0 & f_c \end{bmatrix} \begin{pmatrix} \tan(u^{tilt}_k) \\ \tan(u^{pan}_k) \end{pmatrix}$$

where f_c is the normalized focal length of the visual sensor; u^{tilt}_k, u^{pan}_k are the control signals to the pan/tilt servo in the unit of radian.

The distance between location of each object z_{ki} and the predicted location of the target z_k is employed as the dynamic feature defined by:

$$\tilde{z}_k = z_{ki} - z_k$$

2.2.2 Pattern recognition

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 training data. We assume these features are independent and fulfill normal distribution. Thus we can define the simplified discriminant function and classifier with weightings as:

$$f_1'(\alpha_i) = \sum_{k=1}^{n} (\alpha_{i,k})^2 w_k \quad (1)$$

$$h'(\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots) = \arg\min_i f_1'(\alpha_i) \quad (2)$$

where $\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n})^t$ is derived by normalizing the feature vector of the object i: $[\phi_1,\ \phi_2,\ \phi_3,\ \phi_4,\ d,\ \tilde{z}_k]_i$ based on the distribution of each feature.

Decision making is based on the discriminant function with weightings:

$$D = \begin{cases} \text{target} = h'(\alpha_1, \ldots, \alpha_i, \ldots), & \text{if } \min f_1'(\alpha_i) < \Gamma \\ \text{no target in the image}, & \text{if } \min f_1'(\alpha_i) \geq \Gamma \end{cases}$$

Γ is the threshold valued which is decided experientially based on training data. Then this function of the classifier is to assign the target class to the object whose value of the discriminant function is the smallest, and also smaller than the specified threshold value.

2.2.3 Finite state machine

The finite state machine plays a critical role in our project to dynamically chooses necessary features and give different weightings to each features in the discriminant function under different tracking conditions, shown in Figure 2. In the state 0 ($S0$): since there is no target found in the image, only pure static features are used in discriminant function (Equation 1) to identify the target in the entire image.

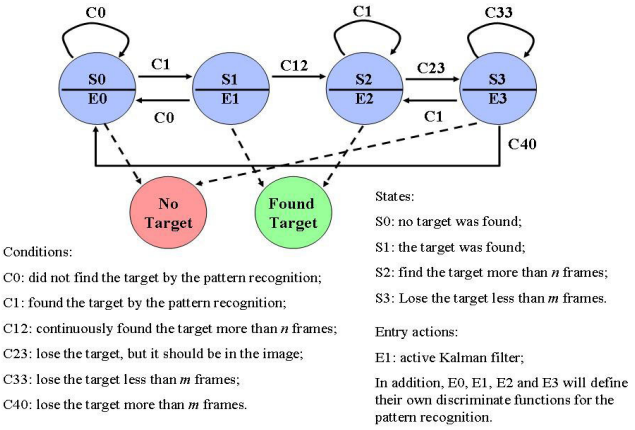

Figure 2: Decision making using finite state machine

In the state 1 ($S1$): the same target has continously been found by the algorithm less than n frames. The discriminant function in (1) still uses static features in the pattern recognition, but enables a Kalman filtering to estimate the possible location of the target in the next frame.

In the state 2 ($S2$): the same target has continously been found by the algorithm more than n frames. We then have confidence to decide that this target is the one we want and use both static and dynamic features in the discriminant function to identify and lock the target in the successive frames, which can reduce the error of detecting a false target.

Figure 3: Tracking with occlusion

In the state 3 ($S3$): the target is lost by the vision detection algorithm. If the partial occlusion detection, based on Chi-square test, indicates that the target is still in the image and may be partially occlude, we manage to give the high weightings to the dynamic features and the color feature, which are less affected by the partial occlusion, while reduce the weightings to the geometric features. We keep this setting of the discriminant function for m frames to try to retrieve the target in the next m frames.

Figure 3 shows an example of the tracking a toy car using the proposed the vision detection algorithm in the ground test. In Figure 3, the solid window is the measured location of the target, and the dash window is the predicted location of the target in the image plane. First the vision detection algorithm automatically initialize the detection, then lock the target. When the target is partially occluded, the vision algorithm give high weightings to the dynamic and color features in the discriminant function. Thus, the target still can be detected, even thought it is partially occluded.

3 Tracking control

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. In our project, the tracking error function is defined in the visual sensor frame as:

$$e(t) = \begin{pmatrix} \theta_c \\ \phi_c \end{pmatrix} - \begin{pmatrix} \theta_c^* \\ \phi_c^* \end{pmatrix} = \begin{pmatrix} \tan^{-1}(\frac{\bar{x}}{f_c}) \\ \tan^{-1}(\frac{\bar{y}}{\sqrt{f_c^2 + \bar{x}}}) \end{pmatrix} - \begin{pmatrix} \theta_c^* \\ \phi_c^* \end{pmatrix} \quad (3)$$

where $[\theta_c,\ \phi_c]^{\mathrm{T}}$ are the measured relative angles between the physical center of the visual sensor and the target illustrated in Figure 4, and $[\theta_c^*,\ \phi_c^*]^{\mathrm{T}}$ are the desired relative angles.

The purpose of the design of the tracking control law is to minimize the error function given in (3) by choosing a suitable control input $u(k)$. We employ a PI controller

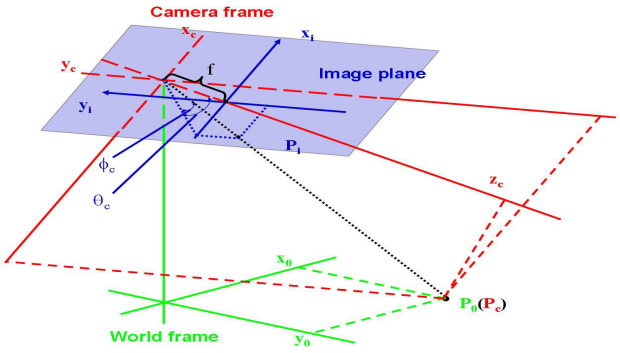

Figure 4: The relative angle between the UAV and target.

Table 1: Experimental results

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%

given by

$$u(k) = k_p e(k) + k_i T_s \sum_{i=1}^{k} e(i) \tag{4}$$

We choose $k_p = 1$ and $k_i = 0.75$ for both of the pan/tilt servo controllers based on the model of the pan/tilt servo mechanism, and verify the controllers in simulation and ground tests.

4 Experiment results and discussion

The proposed vision-based tracking algorithm is implemented in the on-board system of the unmanned helicopter: SheLion. The processing rate of the algorithm is 16 fps. During the real flight tests, the helicopter is manually controlled to hover at a fixed position 10 meters above the flat ground, and the on-board visual tracking system automatically identify and track the ground moving target: a toy car, which is manually controlled to randomly move in the flat ground. We performed nine times of visual tracking tests and the tracking results are shown in table 1. During these tests, the visual tracking system can successfully identify and track the ground target. One example of the tracking errors in vertical and horizontal direction is shown in Figure 5, 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. The tracking errors, however, are greater than these in ground tests, since the ego-motion and vibration of the UAV platform may degrade the performance of the visual tracking system. That is the reason why we are going to consider the ego-motion compensation of the UAV in the further visual tracking algorithm.

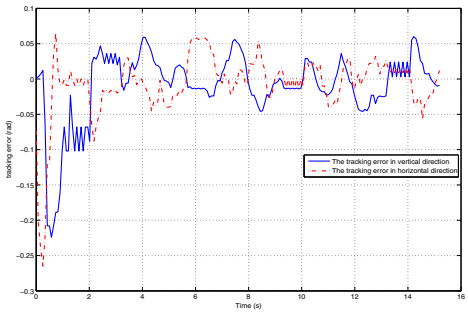

Figure 5: The tracking error of θ_c and ϕ_c.

5 Conclusion

In this paper, we present the design and implement of the visual tracking system to realize on-board tracking a ground moving target for a UAV, which combines the proposed vision-based target detection algorithm with tracking control strategy. The experiment results obtained from the real flight tests of the UAV show that the vision-based tracking system is to be able to automatically identify and track ground moving target, and the tracking error is bounded. To reduce the tracking error, more research effort will be given to integrate the ego-motion of the UAV with the vision-based tracking control system in the further research,.

References

[1] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, pp. 331–340, 2008.

[2] L. Mejias, S. Saripalli, P. Cervera, and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal of Field Robotics*, vol. 23, pp. 185–199, 2006.

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

[4] J. Kim and S. Sukkarieh, "Slam aided gps/ins navigation in gps denied and unknown environments," in *The 2004 International Symposium on GNSS/GPS*, Sydney, Australia, 2004.

[5] E. Trucco and K. Plakas, "Video tracking: A concise survey," *IEEE Transactions on Oceanic Engineering*, vol. 31, pp. 520–529, 2006.

[6] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, 2007.

[7] Q. M. Zhou and J. K. Aggarwalb, "Object tracking in an outdoor environment using fusion of features and cameras," *Image and Vision Computing*, vol. 24, pp. 1244–1255, 2006.

[8] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos, "Robust target detection and tracking through integration of motion, color and geometry," *Computer Vision and Image Understanding*, vol. 103, pp. 121–138, 2006.

[9] A. R. Smith, "Color gamut transform pairs," in *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, New York, USA, 1978, pp. 12–19.

[10] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, pp. 11–32, 1991.