Research on computer vision-based for UAV autonomous landing on a ship

Guili Xu, Yong Zhang*, Shengyu Ji, Yuehua Cheng, Yupeng Tian

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, PR China

1. Introduction

Many autonomous landing systems for Unmanned Aerial Vehicles (UAVs) are based on GPS and a dedicated close range sensor for accurate altitude measurement (radar altimeter, sonar, infrared or theodolites). However, in urban environments buildings and other obstacles disturb the GPS signal and can even cause loss of signal. Therefore, only with independent control of navigation and landing guidance system can meet the requirements during the wartime for a UAV. There are many universities and research institutions engaged in UAV technology of visual navigation (Sharp et al., 2001; Saripalli et al., 2002, 2003; Ettinger et al., 2002). Sharp et al. (2001) and Saripalli et al. (2002, 2003) all choose a special platform for landing a helicopter. Ettinger et al. (2002) extract the horizon from images obtained by the aircraft. Some of these studies are for the autonomous unmanned helicopter, or their algorithm is time-consuming or it requests the images have a high-resolution and high-definition.

For the purpose of implementing UAV’s landing on the deck automatically and accurately under all-weather, a new program of UAV’s automatic landing on the ship’s deck is proposed in this paper, whose main navigation is GPS/SINS combined with the height instrument and assistant navigation is computer vision/SINS combined with the height instrument based on the cooperative object on the runway of the ship. Namely, we set a cooperative object on the runway of the ship, then the cooperative object can be recognized by using computer vision, ultimately the UAV can precisely navigate and land automatically. This technology also resolves the potentially dangerous problem of GPS denial or spoofing by outside sources.

2. System design

The overall landing strategy is best described as follows. Before UAV take off, SINS and GPS start to work at the same time. During flight, the flight control system determines whether the GPS is working normally. If normal, then use GPS/SINS combined with altimeter to navigate; else, the main task of navigation is completed by INS and altimeter. However, due to INS has accumulated error by time, it can only guide the UAV to the approximate landing place. Then we use computer vision to recognize the cooperative object which is put on the runway, and we can get some landing parameters by using the information obtained from the cooperative object. Eventually, UAV can navigate precisely and land automatically. Fig. 1 shows the overall landing plan.

3. Cooperative object detecting using vision

3.1. Cooperative object design

Through the analysis of visible light and infrared radiation, we find visible light images have a high dependence on the light. So when recognizing in the condition of visible light, requires the target not only has a good property of reflection, but also contrasts clearly with the background; but for the infrared radiation images, temperature difference between the target and background is the main factor that affect our recognition (Zhou and Liu, 2004 and...
Therefore, in order to implement UAV’s landing automatically under all-weather, we choose infrared radiation images in our experiment and use a thermal imager to take the pictures (Yakimenko et al., 2000, 2002). Then, we analyze the influence of atmosphere on infrared radiation, in the wave band of 3–5 μm and 8–12 μm, transmission rate has no obvious peak attenuation, so we select these two bands. Shown by Eq. (1), the biggest temperature range of radiation efficiency of 3–5 μm and 8–12 μm is 460.8–950.1 °C and 32.7–185.6 °C. Comparing the two temperature ranges, it is clear that the corresponding temperature range of 8–12 μm wave band can satisfy our requirements better:

\[ \lambda T_e = 3669.73 \text{ (μm K)} \]  

At the same time we opt for a black powder whose emissivity is larger than 0.92 paint on the surface of the cooperative object. It can improve the efficiency of thermal radiation and increase the rate of identification. In this paper, we design the “T” model, Fig. 2 shows our landing target design.

3.2. Image processing

3.2.1. Segmentation

Before recognition of the target, segmentation is an essential and important step. Threshold method is a simple and effective method of image segmentation, in this method, the image will be divided into several parts based on one threshold value or several threshold values, and if they belong to the same part of the threshold value they are in the same object. As threshold method with simple calculation, in the fields which need efficiency, it has been widely used. In this paper, the background and cooperative object have a great difference of gray, so we choose this method. In order to separate more robustly, the system should be able to automat-
ically select the threshold. Next several common and relatively good automatic segmentation threshold methods will be discussed.

(1) Otsu method In 1979, Otsu proposed a common threshold method which is based on the histogram of the image. This is also the most commonly used method, so we will not elaborate here.

(2) Iterative method In this method, a threshold value “T” should be chosen firstly, then segment the image into two parts based on the “T”, after that, from Eqs. (2)–(4) we can get another threshold value, and then compute again and again until the result achieve the stated requirement (Sonka et al., 2003).

\[
\begin{align*}
\mu_B &= \frac{\sum_{(i,j): \text{background}} f(i,j)}{\# \text{background pixels}}, \\
\mu_O &= \frac{\sum_{(i,j): \text{objects}} f(i,j)}{\# \text{objects pixels}}, \\
T &= \frac{1}{2} (u_O + u_B),
\end{align*}
\]

where \( \mu_B \) is the average gray of the background and \( \mu_O \) is the average gray of the object, \( T \) is the new threshold value. Fig. 3 shows the results of these two methods: (a) is original image, (b) and (c) is corresponding result of Otsu method and iterative method.

From the result we find these two methods cannot segment the target from the image, so we choose the following method in our experiment.

(3) Automatic threshold segmentation As the cooperative object has the highest temperature, it is brighter than any other part of the image, meanwhile, the cooperative object occupy a very small percentage in the image, so the gray value of the cooperation object should be distributed in the largest peak histogram. Through our observation of lots of images, we find it is true. Therefore, automatic threshold segmentation method used in this paper includes steps as below:

1. Find the location \( T_0 \) where the largest peak of the histogram.
2. We store the gray value which is greater than \( T_0 \) in the array \( R \), then the proportion of object in the array \( R \) increased significantly, after that, we get threshold value \( T_1 \) through Otsu method.
3. Produce a binary image use the threshold value \( T_1 \).

Take Fig. 4 for example, the first line is the original images; the second line is the histogram of the original images; The third line is images that replace the gray value which is less than or equal to \( T_0 \) by 0, other remain; the fourth line is images that segmented with the threshold value \( T_0 \); the last line is images that segmented using the method in this paper.

Confirmed by a large number of experiments, the images obtained using the above method can meet the requirements of the follow-up treatment.

3.2.2. Extraction of edge information

After the segmentation, we get the pixels integration of the region. If we compute using the pixels within the region directly, the recognition algorithm will be too great. Therefore, in order to reduce the amount of computation, we compute using the edge information. To obtain the edge data of the region, we need image edge detection and tracking. In this article, we select Sobel method for edge detection, and then obtain the edge information through the chain code and segment table method.

3.3. Detecting using affine moment invariants

Moments invariants are features based on statistical moments of the characters, they are traditional and widely-used tool for
character recognition. Classical moment invariants were introduced by Hu and they have numerous successful applications not only in character recognition (Mercimek et al., 2005).

In 1962, Hu derived seven moment invariants that are invariant under translation, rotation and scaling of the object, they are

\[
\begin{align*}
\phi_1 &= \eta_{20} + \eta_{02} , \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11} , \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 , \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 , \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \\
&+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 , \\
\phi_6 &= (\eta_{30} - \eta_{02})(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\
+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) , \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \\
&- (3\eta_{30} - 3\eta_{12})(\eta_{12} + \eta_{03})(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 .
\end{align*}
\]

Affine moment invariants (AMIs) have been derived recently as the tool for recognition of distorted objects. AMIs is moment-based features invariant under general affine transform. The AMIs was derived by means of the theory of algebraic invariants. Full derivation and comprehensive discussion on the properties of invariants can be found in the paper by Flusser and Suk (Flusser et al., 2007; Lemaître et al., 2007).

The AMIs is invariant under general affine transformation:

\[
\begin{align*}
\mathbf{u} &= a_0 + a_1 x + a_2 y , \\
\mathbf{v} &= b_0 + b_1 x + b_2 y ,
\end{align*}
\]

where \((x, y)\) and \((u, v)\) are coordinates in the image plane before and after the transformation, respectively. Four simplest AMIs that we have used for character recognition are listed below:

\[
\begin{align*}
I_1 &= (u_{20}u_{02} - u_{11}^2) / u_{00}^4 , \\
I_2 &= (u_{20}^2u_{02}^2 - 6u_{30}u_{21}u_{12}u_{03} + 4u_{30}u_{12}^2 + 4u_{21}^2u_{03} \\
&- 3u_{21}u_{03}^2) / u_{00}^6 , \\
I_3 &= (u_{20}(u_{21}u_{03} - u_{12}^2) - u_{11}(u_{30}u_{03} - u_{21}u_{12}) \\
&+ u_{02}(u_{10}u_{12} - u_{11}^2)) / u_{00}^3 , \\
I_4 &= (u_{20}^3u_{02}^3 - 6u_{20}u_{02}u_{11}u_{12}u_{03} - 6u_{20}u_{02}u_{21}u_{12}u_{03} + 9u_{20}u_{02}u_{12}^2u_{12} \\
&+ 12u_{02}u_{10}u_{12}u_{12}u_{03} + 6u_{20}u_{11}u_{03} - 6u_{02}u_{11}^2u_{21}u_{12} \\
&+ 9u_{20}u_{10}u_{12}u_{12}u_{03} - 6u_{20}u_{11}u_{02}u_{30}u_{21} \\
&+ u_{02}^3u_{30}^3) / u_{00}^3 .
\end{align*}
\]

4. Angle of yaw

Given the labeled feature points, estimating the UAV state is the so-called model based camera pose estimation problem from computer vision. We only calculate the angle of yaw.

\[
\text{Fig. 6. Different objects.}
\]

\[
\text{Fig. 7. Camera in different pitch angles.}
\]

\[
\text{Fig. 8. Camera in different roll angles.}
\]

\[
\text{Fig. 9. Camera in different distances.}
\]
From Fig. 5, the broken line is the axis of the image, the real line is the axis of the cooperative object, and the angle of these two lines is the angle of yaw.

5. Experiments and discussion

5.1. Experiments

In order to verify the accuracy and reliability of affine invariant moments, we did several experiments.

(1) Different objects, we can see Fig. 6;
(2) When the camera in different pitch angles, we can see Fig. 7;
(3) When the camera in different roll angles, we can see Fig. 8;
(4) When the camera in different distances, we can see Fig. 9;
(5) When the object in different directions, we can see Fig. 10.

We calculate their AMIs (a total of 29 results) and then get a graph from the results, Fig. 11a–d is the corresponding result of $I_1$, $I_2$, $I_3$, $I_4$.

(6) Infrared images of cooperative object

5.2. Discussion

In Fig. 11, we can see the AMI results have a clear distinction between the cooperative object and other objects (1, 2, 3, 4, 6, 7 are results of other objects in Fig. 6), so the target can be recognized easily from the images. At the same time, the four AMIs we used in this paper are very stable, they do not change significantly, so the AMIs method can be used to identify targets.

In our experiments, we get 36 different infrared images (in Fig. 12, we just list 6 of them). We extract the target from the background and then recognize it by using affine moment invariants. The result shows that 35 of the images have been identified correctly, only one missed, the recognition rate is 97.2%.
6. Conclusions

In this paper, we study the basic research concerning automatic UAV navigation and landing on the deck. We analyze the infrared radiation images in our experiments by extracting the target from the background and then recognizing it through the use of affine moment invariants. Based on our experiments, the average recognition time is 17.2 ms and the recognition rate is 97.2%. This type of speed is expected to improve the reliability and real-time performance of autonomous UAV landing. In the near future, we would like to focus our attention to fulfill the pose estimation of the UAV (calculate the angle of pitching and angle of roll and some other parameters).

Acknowledgement

This work was supported by the National Aviation Science Foundation of China under Grant No. 20060752006.

References


