Automatic UAV Positioning with Encoded Sign as Cooperative Target

Xu Zhongxiong[1], *Shao Guiwei*[2], *Wu Liang*[1], *Xie Yuxing*[1], *Ji Zheng*[*1]

1. School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, P. R. China;
2. China Electric Power Research Institute, Wuhan 430079, P. R. China

(Received 24 April 2017; revised 06 June 2017; accepted 30 August 2017)

Abstract: In order to achieve the goal that unmanned aerial vehicle(UAV) automatically positioning during power inspection, a visual positioning method which utilizes encoded sign as cooperative target is proposed. Firstly, we discuss how to design the encoded sign and propose a robust decoding algorithm based on contour. Secondly, the Adaboost algorithm is used to train a classifier which can detect the encoded sign from image. Lastly, the position of UAV can be calculated by using the projective relation between the object points and their corresponding image points. Experiment includes two parts. First, simulated video data is used to verify the feasibility of the proposed method, and the results show that the average absolute error in each direction is below 0.02 m. Second, a video, acquired from an actual UAV flight, is used to calculate the position of UAV. The results show that the calculated trajectory is consistent with the actual flight path. The method runs at a speed of 0.153 s per frame.

Key words: unmanned aerial vehicle(UAV); cooperative target; encoded sign; visual positioning

CLC number: V249.3 **Document code**: A **Article ID**: 1005-1120(2017)06-0669-11

0 Introduction

Unmanned aerial vehicle(UAV) technology has become more and more widely used in most fields[1]. UAVs are distinguished for their ability to fly at various speeds, to stabilize their position, and to hover over a target[2]. These advantages enable them suitable to replace humans in some special tasks when human intervention is dangerous, difficult and expensive. Therefore, UAVs are increasingly used in power inspection.

To maximally reduce the power equipment failures and huge economic losses, transmission line and substation equipment ought to be regularly inspected for detecting defects as soon as possible to arrange the maintenance plan[3]. It is well known that, as an essential part of modern power systems, the stable operation of transmission lines plays a crucial role in the entire power system. Transmission line inspection is a necessary mean for operation and maintenance to ensure the secure function of the power system and afford good service for users[3]. There are two common methods of power equipment inspection, one is foot patrol, and the other is inspection by mobile robot or UAV[4-6]. The foot patrol inspection mainly relies on people to traverse along the scheduled route and inspect the power equipment such as voltage transformer, current transformer, load switch, circuit break and so on. It is inefficient, tedious and laboring. More and more researchers have spotted that and exploited UAV for power inspection thanks to its advantages mentioned above.

During power inspection, UAVs should avoid collisions with power equipment and to achieve the autonomous inspection, UAVs should be able to acquire its position using sensors mounted on the UAV[7]. A common way of providing position is the use of GPS sensor[8], but GPS has two shortcomings: Electromagnetic in-

[*] Corresponding author: E-mail: jz07@whu.edu.cn.

How to cite this article: Xu Zhongxiong, Shao Guiwei, Wu Liang, et al. Automatic UAV positioning with encoded sign as cooperative target[J]. Trans. Nanjing Univ. Aero. Astro., 2017,34(6):669-679.
http://dx.doi.org/10.16356/j.1005-1120.2017.06.669

terference from transmission line may cause poor positioning accuracy; and the accuracy of position only reach 1—2 m based on point positioning of GPS, which cannot effectively avoid dangerous equipments. Therefore, vision based navigation is a wise choice. Monocular camera, a light weight and an inexpensive sensor has been used for completely autonomous flights[9]. A vision based framework[10] estimates the five degrees-of-freedom pose using a camera mounted on a quadrotor helicopter. Real-time position for autonomous navigation of a UAV[11] has been achieved on a mobile processor using an on-board computing unit and multiple sensor (laser, camera, and inertial measaremeni unit (IMU). A scanning laser range sensor retrofitted with mirrors is used as primary source of information for position estimation. This is followed by simultaneous localization and mapping algorithm (SLAM). A landmark-based monocular localization technique[12] uses the time-consuming scale invariant feature transform (SIFT) for absolute pose estimation. Monocular vision[13] has been used to extract edges and compared with known 3D model of the environment followed by particle filter for localization. The autopilot uses an optic flow-based vision system[14] for pose estimation based on autonomous localization and scene mapping. Strelow and Singh[15] proposed a real-time visual-inertial navigation using an iterated extended Kalman filter (EKF), but its complexity grows with the number of features. Mourikis et al. [16] proposed a more efficient approach which considered pairwise images for visual odometry and fused the output with inertial measurements in an EKF. The IMU measurements[17, 18] have been coupled with EKF SLAM framework. However, the computational cost of EKF SLAM is $O(N^2)$ for N features. What we need is to estimate the pose of UAV using a light weight sensor requiring minimal processing.

Considering the special scene of the transmission line, we exploited the encoded sign as cooperative target to achieve visual-based positioning. The encoded sign was hanged on the pylon and monocular camera was mounted on UAV. We calculated the position of UAV by using the projection of more than three encoded signs on the image.

1 Design of Encoded Sign and Its Decoding

In the field of high-precision photogrammetry, it is necessary to set up a certain number of measurement signs as control points, then the geometric parameters can be calculated by measuring the object and image position of the measurement signs[19]. Inspired by photogrammetry, a UAV needs to know the location of the object control point and its location on image. In visual navigation, the artificial sign is usually used as control point or cooperative target[20]. Here, we use encoded signs to distinguish different locations of object points. The encoded sign should be convenient for identifying from the image and decoding fastly.

1.1 Design of encoded sign

Since the rectangle has significant Haar characteristics, and it is convenient to train a classifier by Adaboost algorithm[21], the shape of encoded sign is designed to be circular, and six rectangles enclosed in the circle are used to encode the sign. Among the six rectangles, there are four coded rectangles, one location rectangle and one directional rectangle (Fig. 1). The directional rectangle is to determine the starting position of the decoding so that the encoded sign is of rotation invariance. The function of the location rectangle is to determine the center of the encoded sign. The sign is encoded in binary: The white coded rectangle represents 1, while the black 0. The black coded rectangle is not shown in the sign due to the black background. Since there are four coded rectangles, the total number of combinations are $2^4=16$ (Fig. 2).

1.2 Decoding of encoded sign

A robust decoding algorithm is one of the most critical parts of UAV autonomous positioning. We proposed a decoding algorithm based on

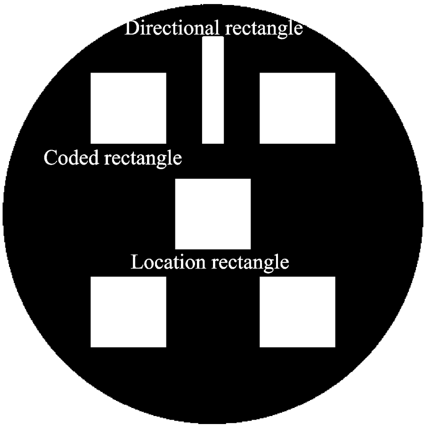


Fig. 1 Composition of encoded sign

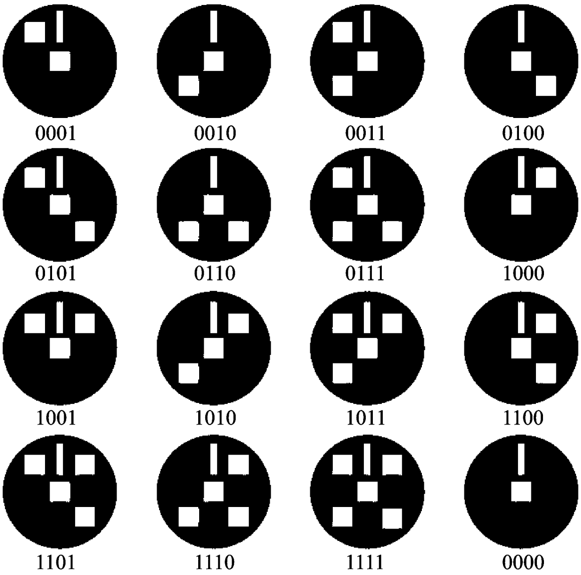


Fig. 2 Coding combinations

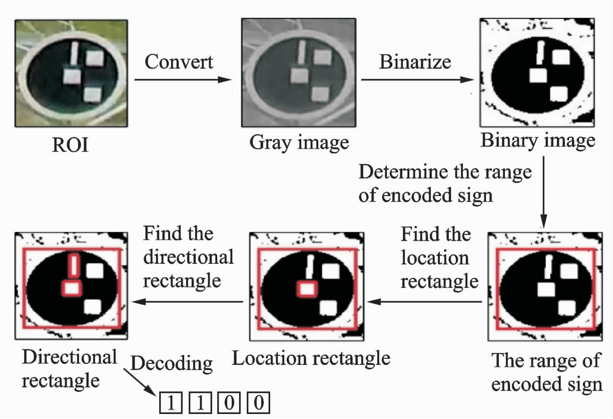

Convert
Binarize
ROI
Gray image
Binary image
Determine the range of encoded sign
Find the directional rectangle
Find the location rectangle
Directional rectangle
Decoding
Location rectangle
The range of encoded sign
1 1 0 0

Fig. 3 Flow chart of decoding

contour. The algorithm maintains strong robustness under some extreme conditions such as blurred image, small image and large angle shooting. The details (Fig. 3) of decoding are as follows:

(1) Extract the region of interest

In the real video, acquired by UAV, each frame may contain multiple encoded signs, so it is necessary to determine the approximate range of each sign in the entire image, as called region of interest (ROI). We exploited a classifier to detect ROI from image frame. The classifier is trained by Adaboost algorithm, it will be discussed in Section 2.

(2) Convert the ROI image from RGB to gray

There are only two colors in encoded sign, black and white, so the ROI image is converted into single-channel gray image for the convenience of decoding.

(3) Binarize the ROI image

Since the decoding algorithm is based on the contour, ROI image needs to be binarized before extracting contour. The ROI image is almost full of encoded sign, and there are only two colors in encoded sign, so the threshold to binarize the ROI image is set to the mean gray of ROI image. Those pixels greater than the threshold are assigned a value of 1, while pixels less than the threshold are assigned a value of 0.

(4) Accurately determine the range of encoded sign

In order to eliminate the interference of complex surrounding, it is necessary to accurately determine the range of encoded sign. First of all, find all contours in the binary image, and then fit the external rectangle for each contour. Since the ROI image is almost full of encoded sign, the largest external rectangle is considered to be the exact range of encoded sign.

(5) Find the location rectangle

The location rectangle is located in the center of encoded sign (Fig. 1). Therefore, traversing all the external rectangles in the range of encoded sign, the external rectangle which is closest to the center of encoded sign is determined as location rectangle.

(6) Find the directional rectangle

The directional rectangle is the smallest in area and the largest in aspect ratio (Fig. 1). Traversing all external rectangles in the range of encoded sign, the external rectangle with smallest area and largest aspect ratio is the directional rectangle.

(7) Decoding

After finding the location rectangle and the directional rectangle, the starting direction is defined as the center of location rectangle point to the center of directional rectangle (Fig. 4). Scanning clockwise from the starting direction, if there is a coded rectangle, the value 1 is assigned to the corresponding binary bit until returns to the starting direction.

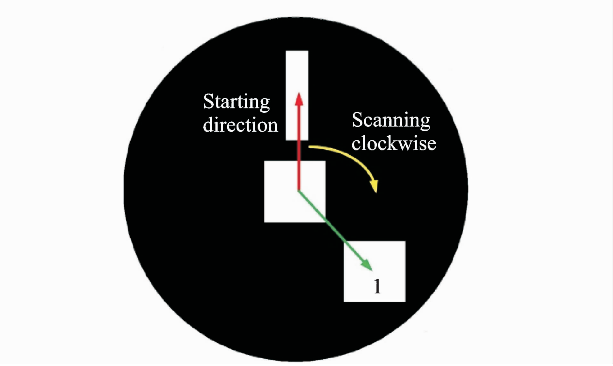


Fig. 4 Decoding

2 Detection of Encoded Sign From Image Frame

Before decoding the encoded sign, the encoded sign should be detected from the image frame. Since UAV needs to calculate its position in real-time, algorithm which can detect the encoded signs in real-time is critical. Inspired by the work of Paul Viola and Michael J. Jones[21], the Adaboost algorithm was used to construct a face detection system which is approximately 15 faster than any previous approach, the face can be detected in real-time. Therefore, we exploited the Adaboost algorithm to train a classifier which can be used to detect the encoded sign from image frame in real-time.

2.1 Classifier training

Before training, the encoded sign should be represented by some features. The encoded sign is consisted of rectangles, so Haar features which have been used by Papagergiou[22] is of value. More specifically, we use three kinds of features. The value of a two-rectangle feature (shapes ① and ② in Fig. 5) is the difference between the sums of the pixels within two rectangular region. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle feature (shapes ③, ④ in Fig. 5) computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a four-rectangle feature (shape ⑤ in Fig. 5) computes the difference between diagonal pairs of rectangles[21]. Note that the number of Haar features is far larger than the number of pixels. For example, given that the base resolution of detector 24 pixel × 24 pixel, the set of rectangle features is quite large, 45 396. Computing all of the features is time consuming.

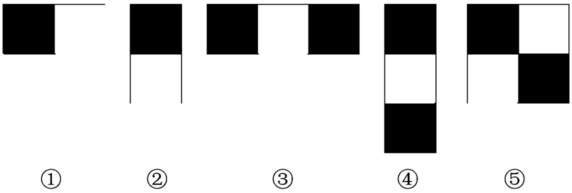


Fig. 5 Haar feature

The Adaboost algorithm is to select a very small number of these features to determine the weak classifiers[21], and then to combine the weak classifiers into a strong classifier. The procedure of Adaboost (Fig. 6 (a)) can be described as follows:

(1) Given example images $(x_1, y_1), \cdots, (x_n, y_n)$ where $y_i = 1$ for positive (the image with the encoded sign) and $y_i = 0$ for negative (the image without the encoded sign) examples. The size of the example is 24 pixel × 24 pixel.

(2) Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives, respectively.

(3) For $t = 1, \cdots, T$:

① Normalize the weights

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

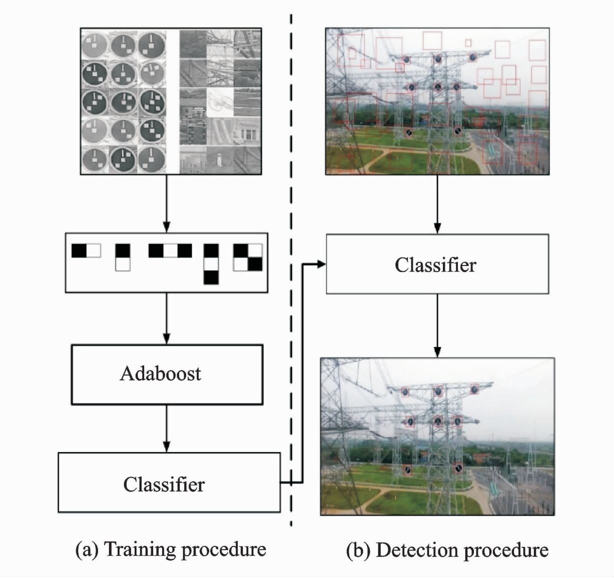


Fig. 6 Training and detection procedure

so that w_t is a probability distribution.

② For each feature j, train a weak classifier h_j which is restricted to a single feature. The error is evaluated with respect to $w_{t,i}$

$$\varepsilon_j = \sum_i w_{t,i} \left| h_j(x_i) - y_i \right|$$

③ Choose the classifier h_t with the lowest error ε_t.

④ Calculate the weight of the classifier h_t

$$\alpha_t = \frac{1}{2}\log\frac{1-\varepsilon_t}{\varepsilon_t}$$

⑤ Update the weights

$$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$.

(4)The final strong classifier

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t\right)$$

2.2 Encoded sign detection

Now the strong classifier can be used to detect the encoded sign. The procedure of detection (Fig. 6 (b)) is as follows:

(1) Initialize the search window

Since the size of the example is 24 pixel×24 pixel, the initial size of the search window is set as 24 pixel×24 pixel.

(2) Traverse the entire image

Slide the search window with a fixed stride, and calculate the Haar features at each window position. Then the Haar features are fed to the strong classifier $H(x)$ to determine if the search window contains the encoded sign or not.

(3) Expand the size of the search window

In order to detect the encoded sign with different scales, the size of the search window should be expand and repeat step (2) until the size of the search window exceeds half of the image size.

(4)Merge the search results

After steps (2),(3), the same size of adjacent search window may detect the same encoded sign (Fig. 7 (a)) and the search window with different sizes may also detect the same encoded sign (Fig. 7 (b)). To eliminate redundancy, these kinds of search windows should be merged. For the case that the search windows with the same size, if the number (N) of search windows within the neighborhood radius (R) is greater than the threshold (T), then the position of the encoded sign is the average position of N search windows. If N is less than T, it means that there is no encoded sign. For the case that the search windows with different sizes, if the overlapping area of the two windows is greater than the given threshold, there is only one encoded sign exist, and the average position and average size of the two search windows are the final detection of the encoded sign. Otherwise, there are two encoded signs.

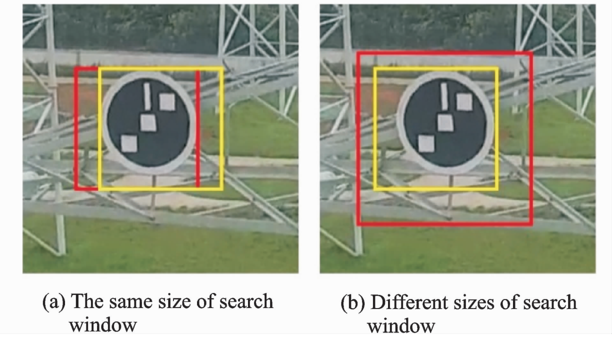

(a) The same size of search window
(b) Different sizes of search window

Fig. 7 Redundancy in two kinds of search window

3 Calculation of UAV Position

Since the camera is fixed on the UAV, we replace the position of the UAV with the camera′s position. Suppose that the camera is located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the z-axis. Such a coordinate system may be called as the camera coordinate frame (Fig. 8). Let the center of the projection be the origin of the camera coordinate frame, and consider the plane $z=f$, which is called as the image plane.

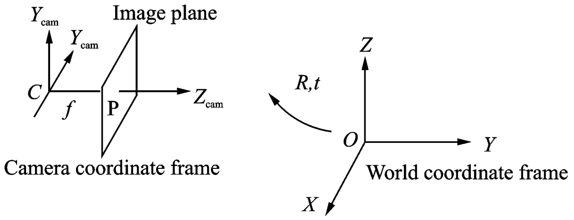


Fig. 8 Transformation between the world and camera coordinate frames

We now introduce the notation $\boldsymbol{X}_{cam}$ for the point in camera coordinate and represented by the homogeneous 4-vector $(X_{cam}, Y_{cam}, Z_{cam}, 1)^T$; $\boldsymbol{x}$ for the image point represented by the homogeneous 3-vector $(x, y, 1)$. Then the mapping between $\boldsymbol{X}_{cam}$ and $\boldsymbol{x}$ can be written in terms of matrix multiplication as:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{pmatrix} \tag{1}$$

where $(p_x, p_y)^T$ are the coordinates of the principle point on the image plane (Fig. 9).

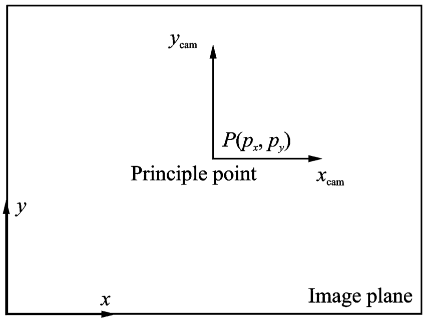


Fig. 9 Image and camera coordinate systems

Now writing

$$\boldsymbol{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{2}$$

Then Eqs. (1) has the concise form

$$\boldsymbol{x} = \boldsymbol{K}[\boldsymbol{I} \mid \boldsymbol{0}]\boldsymbol{X}_{cam} \tag{3}$$

The matrix $\boldsymbol{K}$ is called as the camera matrix, which can be calibrated by the method of Zhengyou Zhang[23].

In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the world coordinate frame. The two coordinate frames are related via a rotation and translation (Fig. 8). If $\boldsymbol{X}$ is an inhomogeneous 3-vector representing the coordinates of a point in the world coordinate frame, and $\boldsymbol{X}_{cam}$ represents the same point in the camera coordinate frame, we may write

$$\boldsymbol{X}_{cam} = \boldsymbol{R}(\boldsymbol{X} - \boldsymbol{C}) \tag{4}$$

where $\boldsymbol{C}$ represents the coordinates of the camera center in the world coordinate frame. In this paper, $\boldsymbol{C}$ represents the position of UAV, and a rotation matrix $\boldsymbol{R}$ 3×3, which determines the pose of UAV. This equation may be written in homogeneous coordinates as

$$\boldsymbol{X}_{cam} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{RC} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{RC} \\ 0 & 1 \end{bmatrix} \boldsymbol{X} \tag{5}$$

Putting this together with Eq. (3) leads to

$$\boldsymbol{x} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X} \tag{6}$$

where from Eqs. (5), $\boldsymbol{t} = -\boldsymbol{RC}$.

In order to calculate the position of UAV, $\boldsymbol{C}$, $\boldsymbol{R}$ and $\boldsymbol{t}$ should be calculated firstly. Note that $[\boldsymbol{R} \mid \boldsymbol{t}]$ matrix has 6 degrees of freedom: 3 for $\boldsymbol{R}$, and 3 for $\boldsymbol{t}$, a pair of image-object points providing two equations. So we need at least three pairs of image-object points to calculate the parameters, but we usually use more than three to improve the accuracy of the solution.

In our work, $\boldsymbol{x}$ is the image coordinate of encoded signs, and its value is provided by the procedure of detection (Section 2). $\boldsymbol{X}$ is the world coordinate of encoded sign, and its value has been determined at the time of installation. The correspondence between $\boldsymbol{x}$ and $\boldsymbol{X}$ is determined by the procedure of decoding (Section 1. 2). Denote $\boldsymbol{m}$ as the image point matrix and $\boldsymbol{M}$ as the world point matrix, then the parameters $\boldsymbol{R}$ and $\boldsymbol{t}$ can be calculated as follows:

(1) Denote $\boldsymbol{m}_c$ as the mean of $\boldsymbol{M}$;

(2) Calculate the intermediate matrix $\boldsymbol{M}_t$

$$\boldsymbol{M}_t = (\boldsymbol{M} - \boldsymbol{m}_c)^{\mathrm{T}} (\boldsymbol{M} - \boldsymbol{m}_c) \tag{7}$$

(3) Singular value decomposition of $\boldsymbol{M}_t$

$$\boldsymbol{M}_t = \boldsymbol{UWV} \tag{8}$$

And

$$\begin{gathered} \boldsymbol{R}_t = \boldsymbol{V} \\ \boldsymbol{T}_t = -\boldsymbol{m}_c \boldsymbol{R}_t \\ \boldsymbol{M}_{xy} = \boldsymbol{VM}^T + \boldsymbol{T}_t \end{gathered} \tag{9}$$

(4) Find homography between $\boldsymbol{m}$ and $\boldsymbol{M}_{xy}$ by the robust estimation method[24], and the homography matrix can be written as

$$\boldsymbol{H} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{t}'] \tag{10}$$

Normalize every column of $\boldsymbol{H}$

$$\begin{gathered} \boldsymbol{h}_1 = \frac{\boldsymbol{h}_1}{\| \boldsymbol{h}_1 \|} \quad \boldsymbol{h}_2 = \frac{\boldsymbol{h}_2}{\| \boldsymbol{h}_2 \|} \\ \boldsymbol{h}_3 = \boldsymbol{h}_1 \times \boldsymbol{h}_2 \quad \boldsymbol{t}' = \frac{\boldsymbol{t}'}{\| \boldsymbol{h}_1 \| + \| \boldsymbol{h}_2 \|} \\ \boldsymbol{H} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3] \end{gathered} \tag{11}$$

(5) Calculate the rotation matrix and translation

$$\begin{gathered} \boldsymbol{R} = \boldsymbol{HR}_t \\ \boldsymbol{t} = \boldsymbol{HT}_t + \boldsymbol{t}' \end{gathered} \tag{12}$$

Finally, the position of UAV $\boldsymbol{C}$ can be calculated

$$\boldsymbol{C} = -\boldsymbol{R}^{-1} \boldsymbol{t} \tag{13}$$

The position of UAV may be inaccurate due to the wrong decoding of encoded signs. There are three methods in this paper to improve the robustness of calculating the position of UAV. First, coding combination is limited, so the decoding algorithm automatically eliminates the decoding results that do not exist. Second, for the case that decoding result exist but the result is wrong, we use the RANSAC[25] algorithm to cope Ref. with this kind of mistakes in the procedure of finding homography[24]. Third, the positioning results of adjacent two-frame image should be in a short distance, and the current positioning result will be removed if it is far from the previous positioning result.

4 Experiment and Discussions

4. 1 Data

We exploited two kinds of data to complete the experiment, the simulated data and the real data. The former was used to verify the feasibility of proposed method, while the latter was to calculate the position of the actual flight of UAV.

We used Unity3D to generate simulated data. The simulated scene had all the elements we needed(Fig. 10). The image size is 1 920 pixel × 1 080 pixel, and we generated 210 simulated images.

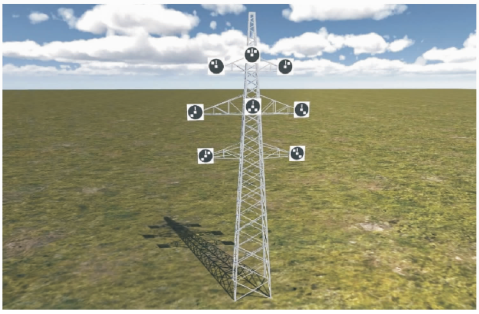


Fig. 10 Simulated scene

The real data was obtained by UAV from the experimental site of China Electric Power Research Institute, Wuhan. Ten encoded signs, with the diameter of 80 cm, were hanged on the pylon (Fig. 11). The focal length of the camera was 5 mm. The image size was 2 530 pixel × 2 000 pixel.

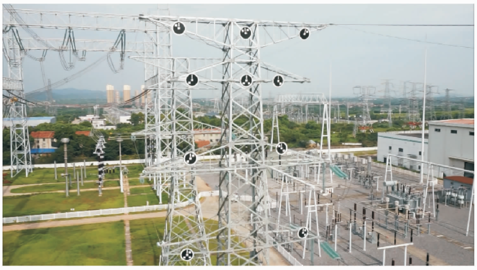


Fig. 11 Real scene

4. 2 Results of detection

In Section 2. 2, the neighborhood radius R was set to the half of the search window. Within the neighborhood radius, the number threshold T was set as 2. We randomly selected 373 images from the real data for detection experiment. The detection results are shown in Table 1.

Table 1 Results of detection

Actual total number (N)	Total number detected (S)	True (T)	False (F)
2 143	2 022	2 019	3

Note: Precision=T/S=99.85% Recall=S/N=94.35%

There were 2 143 encoded signs among 373 images. The classifier deteced 2 022 ROIs which the classifier determined the presence of encoded sign. Among the 2 022 ROIs, there were 3 ROIs did not contain the encoded signs. So the precision was 2 019/2 022=99.85%, and the recall is 2 022/2 143=94.35%. It means that for each frame, the classifier can detect sufficient number of encoded sign for calculating the position of UAV.

4.3 Results of decoding

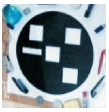
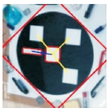
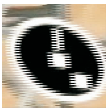
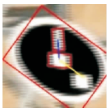
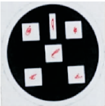
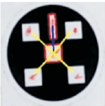
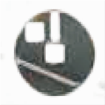
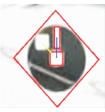
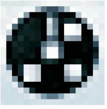
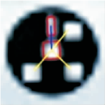
For the 2 019 ROIs which contained the encoded sign, the decoding algorithm mentioned in Section 1.2 was used to decode the encoded sign in each ROI. Results can be seen from Table 2. It shows that the decoding algorithm can decode ROIs with high decoding accuracy.

Table 2 Results of decoding

Code	Total number	Number of errors	Error rate(%)
0001	147	0	0
0010	182	0	0
0100	170	0	0
0101	268	0	0
0110	174	0	0
1000	139	0	0
1001	153	0	0
1010	277	0	0
1100	267	1	0.374 5
1110	242	0	0
Total	2 019	1	0.374 5

In addition, we also verified the robustness of the decoding algorithm. In actual flight, the encoded sign in the image may be deformed and blurred due to UAV vibration and other factors. Moreover, the complex surrounding also brings challenges to the decoding of the encoded sign. We verified our decoding algorithm in different conditions, such as complex surrounding, blurred image by the vibration of UAV, dirt on the surface of encoded sign, small encoded sign and deformed image due to the angle of shooting. Table 3 shows the results of validation. It can be seen that our decoding algorithm has high robustness, and enables the UAV to achieve accurate positioning in complex environment.

Table 3 Robustness test of decoding

ROI	Disturbance	Visualization	Results
	Complex surrounding		1110
	Blurred image		0100
	Dirt on the surface of encoded sign		1111
	Power line passes through the encoded sign		0001
	Small encoded sign (18 pixel×18 pixel)		1110
	Deformed due to the angle of shooting		0100

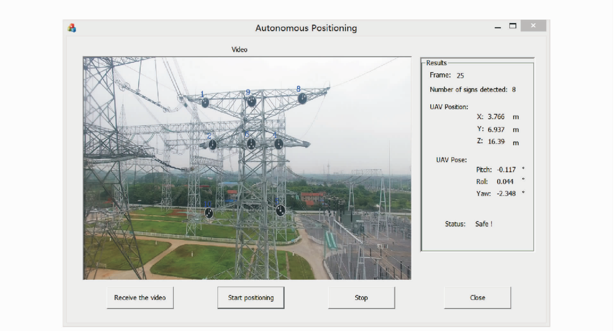
4.4 UAV positioning results

Since the UAV is not equipped with positioning device, the real position of the UAV is unknown. In order to verify the positioning accuracy and the feasibility of our method, we used Unity3D to build a simulated scene (Fig. 10). In the simulated scene, we set the UAV flight path in advance, and the position of each image frame was known. We took 210 images in the simulated scene to verify our method. The positioning results are shown in Table 4. From the results we can see that the average absolute error in each direction is below 0.02 m, thus verifying the feasibility of our method.

Table 4 The results of simulation m

Frame	Actual position			Calculated position			Absolute error		
	X_A	Y_A	Z_A	X_C	Y_C	Z_C	ΔX	ΔY	ΔZ
1	2.552	0.850	26.955	2.550	0.833	26.952	0.002	0.017	0.003
2	2.810	5.419	25.981	2.814	5.432	25.976	0.004	0.013	0.005
3	0.005	3.254	25.539	0.003	3.250	25.538	0.002	0.004	0.001
…	…	…	…	…	…	…	…	…	…
208	2.382	4.428	25.764	2.385	4.422	25.764	0.003	0.006	0.000
209	4.838	2.414	29.406	4.833	2.382	29.407	0.005	0.032	0.001
210	1.931	0.917	27.714	1.908	0.906	27.712	0.023	0.011	0.002

At last, the video, acquired in actual UAV flight, was used to calculate the position of UAV. Fig. 12 is the User Interface of autonomous positioning and Fig. 13 shows the trajectory which is composed by the position of each frame. The blue dot indicates the beginning of the video, while the dark red dot indicates the end of the video. The UAV climbed up at the beginning, then hovered at the top for a short time, and finally flew away from the pylon and slowly declined to the ground. This is consistent with the actual flight path.



Results
Frame: 25
Number of signs detected: 8
UAV Position:
X: 3.766 m
Y: 6.937 m
Z: 16.39 m
UAV Pose:
Pitch: -0.117 °
Roll: 0.044 °
Yaw: -2.348 °
Status: Safe !

Fig. 12 Interface of autonomous positioning

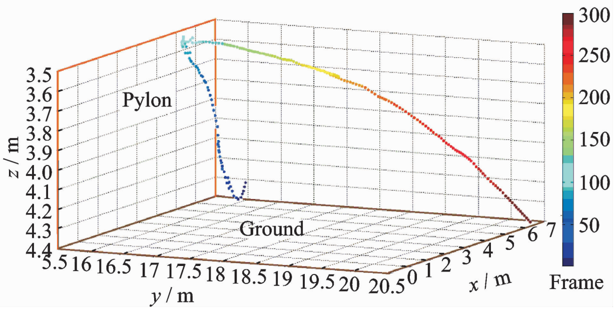


Fig. 13 Trajectory of each frame

4.5 Efficiency of the proposed method

Our method was implemented by C++ language combined with OpenCV, and run in Windows 8.1, Intel core i7 processor clocked at 2.6 GHz alongside 8 GB of memory. The image size is 1 920 pixel×1 080 pixel, and the running time of each module is shown in Table 5. As can be seen, the total processing time for each frame is 153 ms, thus the UAV can avoid the obstacles in time during the transmission line inspection.

Table 5 Running time of each module

Module	Time cost/ms
Detection	150
Decoding	2
Positioning	1
Total time	153

5 Conclusions

A method for autonomous positioning of UAV during the inspection of transmission line was proposed. It provides high-precision obstacle avoidance information for UAV flight control module to avoid dangerous accidents, for example, collision of UAV with electrical equipment. The proposed method improves the automation of transmission line inspection, and reduce the labor intensity. In our future work, deep learning method may be used to avoid obstacles without exploiting encoded signs as cooperative target.

Acknowledgments

This work was supported by the National Key Research Projects (No. 2016YFB0501403), and the National Demonstration Center for Experimental Remote Sensing & Information Engineering (Wuhan University).

References:

[1] HUANG D,XU C, HAN W, et al. UAV velocity

measurement for ground moving target [J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2015, 32(1): 9-15.

[2] KANELLAKIS C, NIKOLAKOPOULOS G. Survey on computer vision for UAVs: Current developments and trends [J]. Journal of Intelligent & Robotic Systems, 2017,87:1-28.

[3] LU S, ZHANG Y, SU J. Mobile robot for power substation inspection: a survey [J]. IEEE/CAA Journal of Automatica Sinica, 2017,99: 1-18.

[4] KATRASNIK J, PERNUS F, LIKAR B. A survey of mobile robots for distribution power line inspection [J]. IEEE Transactions on Power Delivery, 2010, 25(1): 485-493.

[5] SONG Y, WANG H, ZHANG J. A vision-based broken strand detection method for a power-line maintenance robot [J]. IEEE Transactions on Power Delivery, 2014, 29(5): 2154-2161.

[6] TIAN F, WANG Y, ZHU L. Power line recognition and tracking method for uavs inspection [C]// International Conference on Information and Automation. Lijiang:IEEE,2015: 2136-2141.

[7] KOTHARI N, GUPTA M, VACHHANI L, et al. Pose estimation for an autonomous vehicle using monocular vision [C]// Indian Control Conference. Guwahati:IEEE,2017: 424-431.

[8] GUO R, XIAO P, HAN L. GPS and DR integration for robot navigation in substation environments [C]// International Conference on Information and Automation. Harbin:IEEE,2010:2009-2012.

[9] BLOSCH M, WEISS S, SCARAMUZZA D, et al. Vision Based MAV Navigation in Unknown and Unstructured Environments [C]// IEEE International Conference on Robotics and Automation. Anchorage :IEEE,2010:21-28.

[10] EBERLI D, SCARAMUZZA D, WEISS S, et al. Vision based position control for mavs using one single circular landmark [J]. Journal of Intelligent & Robotic Systems, 2011, 61: 495-512.

[11] SHEN S, MICHAEL N, KUMAR V. Autonomous Multi—Floor Indoor Navigation with a Computationally Constrained MAV [C]// IEEE International Conference on Robotics and Automation. Shanghai: IEEE,2011:20-25.

[12] WENDEL A, IRSCHARA A, BISCHOF H. Natural landmark-based monocular localization for MAVs [C]// IEEE International Conference on Robotics and Automation. Shanghai:IEEE,2011:5792-5799.

[13] BUYVAL A, GAVRILENKOV M. Vision-based pose estimation for indoor navigation of unmanned micro aerial vehicle based on the 3D model of environment [C]// International Conference on Mechanical Engineering, Automation and Control Systems. Tomsk:IEEE, 2015:1-4.

[14] KENDOUL F, FANTONI I, NONAMI K. Optic flow-based vision system for autonomous 3d localization and control of small aerial vehicles [J]. Robotics and Autonomous Systems, 2009, 57(6): 591-602.

[15] STRELOW D, SINGH S. Motion estimation from image and inertial measurements [J]. The International Journal of Robotics Research, 2004, 25(2): 1157-1195.

[16] MOURIKIS A I, TRAWNY N, ROUMELIOTIS S I, et al. Vision-aided inertial navigation for spacecraft entry, descent, and landing [J]. IEEE Transactions on Robotics, 2009, 25(2): 264-280.

[17] PINIES P, LUPTON T, SUKKARIEH S, et al. Inertial aiding of inverse depth slam using a monocular camera [C]// IEEE International Conference on Robotics and Automation. Roma: IEEE, 2007: 2797-2802.

[18] BAILEY T, NIETO J, GUIVANT J, et al. Consistency of the EKF-SLAM algorithm [C]// IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing:IEEE,2006:2562-3568.

[19] YONGJUN Z, WANG B, CHEN Q. Automatic extraction algorithm of mark centers in close-range photogrammetry [J]. Journal Of Tongji University (Natural Science), 2014, 42(8): 1261-1266.

[20] HAO X Y, ZHANG Z J, LIU S L, et al. An approach of indoor vision navigation with artificial marks [J]. Journal of Navigation and Positioning, 2013, 1(4): 26-30.

[21] VIOLA P, JONES M J. Rapid object detection using a boosted cascade of simple features [J]. Conference on Computer Vision and Pattern Recognition, 2001, 1: 511-518.

[22] PAPAGEORGIOU C, OREN M, POGGIO T. A general framework for object detection [C]// International Conference on Computer Vision. Bombay: IEEE, 1998:555-562.

[23] ZHANG Z. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations [C]// Proceedings of the Seventh IEEE International Conference on Computer Vision. Kerkyra:IEEE, 1999:666-673.

[24] HARTLEY R, ZISSERMAN A. Multiple View Geometry in Computer Vision[M]. Cambridge: Cambridge University Press, 2003: 116-125.

[25] FISCHLER M A, BOLLES R C. Random sample consensus: a paradigm for model fitting with applica-

tions to image analysis and automated cartography [M]. ACM, 1981: 381-395.

Mr. **Xu Zhongxiong** is a postgraduate student of School of Remote Sensing and Information Engineering, Wuhan University, and his research is focused on computer vision and photogrammetry.

Dr. **Shao Guiwei** is a senior engineer of China Electric Power Research Institute, and his research is focused on intelligent power inspection.

Mr. **Wu Liang** is a postgraduate student of School of Remote Sensing and Information Engineering, Wuhan University, and his research is focused on computer vision.

Mr. **Xie Yuxing** is a postgraduate student of School of Remote Sensing and Information Engineering, Wuhan University, and his research is focused on virtual reality and 3D visualization.

Prof. **Ji Zheng** is an associate professor of School of Remote Sensing and Information Engineering, Wuhan University. His research is focused on photogrammetry, remote sensing and computer vision.

(Executive Editor: Zhang Bei)