TARGET POSITIONING AND TRACKING METHOD OF THE AIRBORNE TRACKER BASED ON MULTI-INFORMATION FUSION

Chuan-Wei ZHANG[1], Li-Ming CHEN[2], Wang RUI[3], Hong-Jun ZENG[4], Da-Ming XU[5]

By fusing the relative position information and the absolute coordinate data, the target positioning and tracking method of the airborne equipment based on multi-information fusion is developed in the present paper. By combining the depth image and color image data, feature points are extracted and matched to the identified targets, so the target relative position to the quadrotor is determined. The absolute coordinates of the quadrotor are derived through the merge of visual and inertial navigations. The target final coordinate information is transmitted to the quadrotor control panel to realize real-time tracking.

Keywords: Image tracking; Integrated navigation; Target location; Multi-information fusion;

Nomenclature

$I^{(n)}$	The grayscale representation of the original image (w × h).
$I(x,y)$	The grayscale value of the image *I* location (x,y), where x and y are two pixel coordinates of image point.
G	Spatial gradient matrix.
b	Mismatched vector.
$\delta I_k(x,y)$	Image difference of k^{th}.
$\mathbf{d}_{\mathbf{opt}}$	Optimal optical flow vector.
G	Spatial gradient matrix.
E(d)	The minimum residual.
$\mathbf{g}^k$	Guess for level k.

[1] College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China, email: czhang26@126.com
[2] College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China
[3] College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China
[4] College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China
[5] College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China

f	Frame per Second.
$\mathbf{V}$	The velocity matrix of the quadrotor.
$\mathbf{V}_{\tilde{n}}$	The velocity matrix corrected the errors.
$\boldsymbol{V}_t$	The velocity matrix obtained by visual navigation to the ground coordinate system.
φ	Roll angle.
θ	Pitch angle.
ψ	Yew angle.
χ	Azimuth of the track angle, The angle between the tangent of the track and the baseline.
γ	Inclination of the track angle. The angle between the flight velocity and the ground plane.
F	Fourier transform.
F^{*}	Complex conjugate of Fourier.
K	The circulant kernel matrix.
γ^{xx}	The first row of K.
γ^{xz}	The z row of K.
$\mathbf{a}_t$	The acceleration of optical flow.
$f''(z)$	Acceleration in the center of the target in the depth image.
R_t	Tracking result.

1. Introduction

In modern life, the application of quadrotors is becoming increasingly extensive. They are suitable for missions such as surveillance and sustained tracking in the near-surface environment. However, manpower is required in many tracking tasks, which results in the inefficiency.

Our research of target positioning and tracking method consist of three parts. First, combination of visual and IMU to get the position of quadrotor. Second, get the position of target by fusing depth images and ‘Red, Green, Blue’ RGB image. Third, by fusing the position of quadrotor and target to get the 3D coordinates of the target.

The traditional on-board target tracking systems use monocular vision method such as ‘Kernelized Correlation Filters’ (KCF) etc. The KCF tracker uses the "kernel trick" approach to extend the correlation filter and is therefore very fast in RGB image tracking. However, this tracker has low accuracy and poor robustness. The scale variation and fast motion lead to target mismatch. So it is difficult to accomplish a series of tracking tasks under complicated environment. Depth information can potentially be exploited to boost the performance of traditional object tracking algorithms. Research into combining depth and color data for tracking is still in its infancy [1].

The relative velocity and acceleration of the target are reflected as the change rate of the target center point in the depth image. In this paper, the image tracking method based on multi-information fusion is proposed by combining depth image and the RGB image information. The method can be taken as a constraint of the KCF tracker by suppressing the target mismatch due to scale variation and fast motion. This method improves the accuracy of image tracking and enlarges the detection range of the system. The absolute coordinates of the quadrotor are derived through the merge of visual and inertial navigations. The target final coordinate information is transmitted to the quadrotor control panel to realize real-time tracking. Eventually, real-time tracking and positioning are realized through the above methods.

2. Combined Visual and inertial navigation

This method first extracts the corner features of the image, By reconstructing the sequence optical flow field of the image, the two-dimensional coordinates of the feature points of the previous frame image are matched and the real-time height data collected by the sonar sensor are fused to obtain the three-dimensional coordinates of the feature points. The moving velocity vector of the quadrotor movement is obtained by differentiating the feature points of two adjacent frames. At the same time, fusing inertial navigation data, integrating, and compensating the errors of visual navigation, and finally realizing the quadrotor trajectory reconstruction. Thus, making the quadrotor stable and autonomous flight [2].

2.1 Pyramid L-K optical flow reconstruction algorithm

The general optical flow algorithm is only suitable for short-distance and slow-moving applications. However, the movement of the quadrotor is fast and long-distance. Therefore, an image pyramid is introduced here to solve this problem and minimize the possibility of not satisfying the motion assumptions, So, as to accurately reconstruct fast and long-distance sports [3].

Let $I^{(n)}$ be the grayscale representation of the original image (w × h), which is located at the lowest level of the pyramid and generates the first, second, ..., n-1 layers of the n-level Gaussian image pyramid [4], denoted $I^1, I^2, \ldots, I^{(n-1)}$.

In $I(x,y)$, $J(x,y)$, the corresponding regional window of $(2w_x+1, 2w_y+1)$ is established respectively, and the pixel in the corresponding window has the same displacement vector $\boldsymbol{d}$ [5].

Definition of spatial gradient matrix $\mathbf{G}$:

$$\mathbf{G} = \sum_{x=u_x-w_x}^{u_z+w_x} \sum_{y=u_x-w_x}^{u_y+w_x} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y^2(x,y) \end{bmatrix}. \quad (1)$$

And suppose mismatched vector $\mathbf{b}$:

$$\mathbf{b} = \sum_{x=u_x-w_x}^{u_z+w_x} \sum_{y=u_x-w_x}^{u_y+w_x} \begin{bmatrix} \delta I_k(x,y)I_x(x,y) \\ \delta I_k(x,y)I_y(x,y) \end{bmatrix}. \quad (2)$$

The grayscale image differential:

$$I_k(x,y) = I^k(x,y) - J^k(x,y). \quad (3)$$

The solution of the optimal optical flow vector $\mathbf{d_{opt}}$ is:

$$\mathbf{d_{opt}} = \mathbf{G^{-1}b}. \quad (4)$$

Iteration of this process by a recursive matrix [6, 7], let $\eta^k = G^{-1}b^k$, k for the *kth* iteration of the estimated translation can get the minimum residual E (d) [8]:

$$\mathbf{E(d)} = \begin{cases} \mathbf{d^0} = \mathbf{0} \\ \mathbf{d}^k = \mathbf{d}^{k\text{-}1} + \boldsymbol{\eta}^k, k \geq 1 \end{cases}. \quad (5)$$

And get the optical flow calculation transfer process:

$$\mathbf{g}^{k-1} = 2(\mathbf{g}^k + \mathbf{d}^k). \quad (6)$$

Finally get the optical flow vector $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$, the corresponding feature point on the image J: $p = u + d$, that is to get in the image J and the image I point $\mathbf{u} = [u_x \quad u_y]^{\boldsymbol{T}}$, Matches a point $\mathbf{p} = [p_x \quad p_y]^{\boldsymbol{T}}$.

2.2 Calculation of velocity

The size and the actual size in the image satisfy the formula (7), ζ is the actual size, σ is the size in the image, h is the distance, B is the magnification,

$$\zeta = \sigma h / B. \tag{7}$$

The real-time height of the point u on the image I is h_u and the real-time height of the point p on the image J is h_p, so we get the actual feature point coordinate matrix $\mathbf{u}_s$ and $\mathbf{p}_s$

$$\begin{cases} \mathbf{u}_s = \begin{bmatrix} \dfrac{u_x \sigma h_u}{B} & \dfrac{u_y \sigma h_u}{B} & h_u \end{bmatrix}^T \\ \mathbf{p}_s = \begin{bmatrix} \dfrac{p_x \sigma h_p}{B} & \dfrac{p_y \sigma h_p}{B} & h_p \end{bmatrix}^T \end{cases}. \tag{8}$$

The frame rate is *fps*, the velocity vector is:

$$\mathbf{V}_i = \begin{bmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix} = fps \cdot \left(\mathbf{p}_{si} - \mathbf{u}_{si} \right). \tag{9}$$

In this system, the threshold value of feature points is set to 500. Because different feature points in the same frame have different directions and sizes of velocity vectors, navigation data may be unstable. Therefore, according to the superposition principle of vectors, the velocity vectors of all the feature points at the same time are superposed to get a velocity vector of the composite motion. At the moment t, the three-dimensional velocity matrix $\mathbf{V}$ of the quadrotor is obtained using the following function: (10)

$$\mathbf{V} = \frac{1}{\boldsymbol{i}} \sum \mathbf{V}_i = \frac{1}{\boldsymbol{i}} \begin{bmatrix} \sum v_{xi} \\ \sum v_{yi} \\ \sum v_{zi} \end{bmatrix} = \begin{bmatrix} v_{xt} \\ v_{yt} \\ v_{zt} \end{bmatrix}, \boldsymbol{i} \le 500. \tag{10}$$

2.3 Data Fusion

On the quadrotor, the visual navigation will produce unavoidable errors due to vibrations of instability. Therefore, the velocity matrix $\mathbf{V}_{ñ}$ is corrected the errors by fusing the IMU data and the visual navigation data. φ , θ and ψ are the roll angle, pitch angle and yaw angle respectively. Here, the RGB image can only get the optical flow of the quadrotor in the projection plane of the camera. To reconstruct the three-dimensional motion of the quadrotor, the coordinates of the feature point in the depth image v_{zt},

$$\mathbf{V}_{ñ} = \begin{bmatrix} v_{ñx} \\ v_{ñy} \\ v_{ñz} \end{bmatrix} = \begin{bmatrix} v_{xt} * (1 - cos\theta) \\ v_{yt} * (1 - cos\varphi) \\ v_{zt} * (1 - cos\psi) \end{bmatrix}. \quad (11)$$

The track angle χ and γ can be obtained from the velocity matrix $\mathbf{V}_{ñ}$,

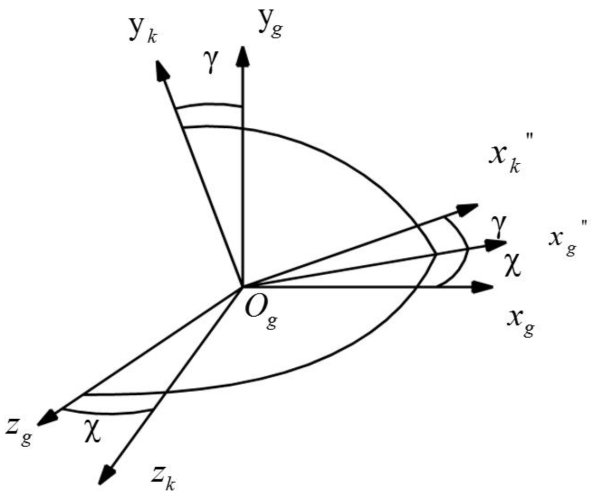

Fig. 1 Definition of Track Angle

As Fig.1, track angle describes the relationship between quadrotor flight trajectory and ground coordinate system.

$$\begin{cases} \chi = arctan \dfrac{v_{ñy}}{v_{ñx}} \\ \gamma = arctan \dfrac{v_{ñz}}{v_{ñx}} \end{cases}. \quad (12)$$

Transforming the velocity matrix $\mathbf{V}_{ñ}$ obtained by visual navigation to the ground coordinate system $\boldsymbol{V}_t$:

$$\boldsymbol{V}_t = \begin{bmatrix} v_{tx} \\ v_{ty} \\ v_{tz} \end{bmatrix} = \begin{bmatrix} v_{\tilde{n}x} \cos\gamma \cos\chi \\ v_{\tilde{n}y} \sin\gamma \\ -v_{\tilde{n}z} \cos\gamma \sin\chi \end{bmatrix}. \quad (13)$$

Integrate $\boldsymbol{V}_t$, and get the displacement vector $\mathbf{S}$ of the quadrotor.

$$\mathbf{S} = \int (v_{tx}, v_{ty}, v_{tz}) dt \ . \quad (14)$$

The displacement vector is decomposed into S_x , S_y and S_z ,finally the displacement matrix $\mathbf{X}$ is obtained.

$$\mathbf{X} = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix}. \quad (15)$$

Finally, the integrated navigation trajectory reconstruction of the quadrotor represents in Fig. 2.

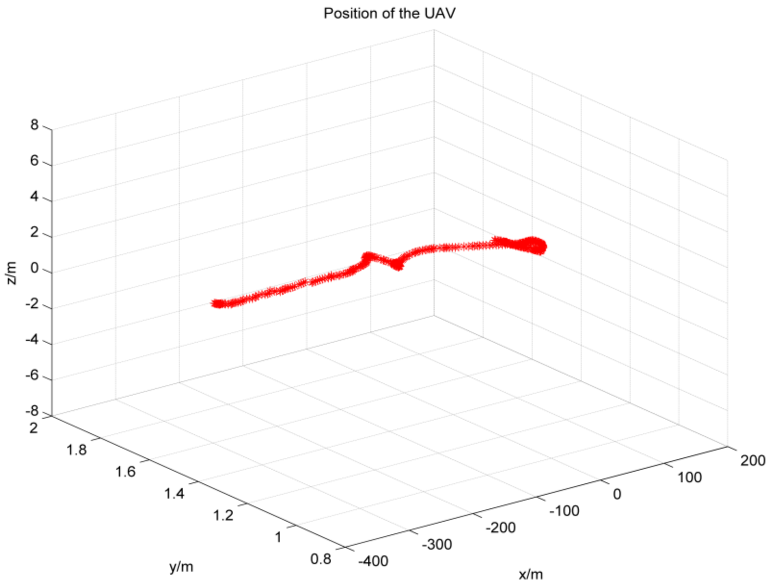

Fig. 2 Integrated Navigation Trajectory Reconstruction

3. Multi-information fusion target tracking

Based on the KCF tracker, a depth image is added to pre-judge the matching result [9]. First, the geometric center of the target is obtained through RGB image tracking, and the position coordinates of the geometric center in the depth image is differentiated to obtain the rate of change of the geometric center on the depth image, which is entered into a filter for evaluation to arrive at an optimal solution that reduces the mismatch of the target tracking.

3.1 KCF image tracking

The KCF tracker uses the "kernel trick" approach to extend the correlation filter and is therefore very fast in RGB image tracking. Additionally, refer the works presented in some trackers for a comprehensive overview of correlation filters [10-12]. As shown in Fig. 3, the KCF tracker includes three steps: training, testing, and retraining [13]. This last step includes updating the model parameters of the previous frame with the interpolation factor of the current frame. KCF uses the properties of a circulant matrix for efficient learning. A $m \times m$ cyclic matrix $\mathbf{C(x)}$ can be constructed by using the m × 1 vector x of the cyclic shift operator.

$$\mathbf{C(x)} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_m \\ x_m & x_1 & x_2 & \cdots & x_{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}. \tag{16}$$

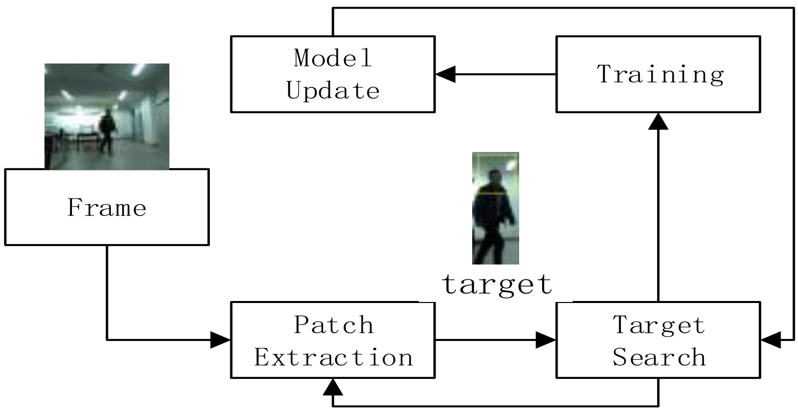

Fig.3 KCF Process

All circular matrices can be diagonalized in a Fourier space using a discrete Fourier matrix: $X = F\ diag(\hat{x})F^{H} X^{H}$ and X into the ridge regression formula to get $w = (X^{H} X + \lambda I)^{-1} X^{T} y$, The Fourier transform of $F(w) = \frac{F^{*}(x) \odot F(y)}{F^{*}(x) \odot F(y) + \lambda}$, where the semicolon is the dot division operation. In this way [14], the dot product operation of the vector can be used to replace the matrix operation, especially the inverse operation, which greatly improves the calculation speed.

Ridge regression in kernel space can be written as: $\alpha = (K + \lambda I)^{-1} y$, In (KCF), this equation can be written as: $\alpha = F^{-1} \frac{F(y)}{F^{*}\gamma^{xx} + \lambda}$, Here γ^{xx} is the column vector of the kernel matrix circular matrix K.

After calculating the sample set α to be classified, an effective estimation can be made using the properties of the circulant matrix and a real-time response $f(z)$ can be made to the image block z [15]:

$$f(z) = F^{-1}(F^{*}(\gamma^{xz}) \odot F(\alpha)). \quad (17)$$

Here a Gaussian kernel function is used to diagonalize the matrix x so that it becomes a kernel matrix cyclic matrix K [16].

$$\gamma^{xz} = \exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|z\|^2 - 2F^{-1}\left(F(x) \odot F^{*}(z)\right)\right)\right). \quad (18)$$

3.2 The depth image tracking method

The KCF tracker is based on a simple chain of processes that estimates an image block at the target location and applies a pre-computed cosine window on the image block to reduce the noise of the Fourier transform. By maximizing the response $f(z)$ to detect the target position and using the Gaussian kernel function to diagonalize the matrix, the ridge regression function is used to reduce the computational complexity, so the tracking frame rate has been greatly improved, which can reach more than 100 *fps*, fully meet the real-time requirements of the quadrotor. However, because of its reliance on a cyclic matrix, its tracking effect is not satisfactory for multi-scale targets and fast-moving targets. Therefore, depth images are used to fuse RGB images to solve the target loss problem of KCF in airborne target tracking [17]. The detailed implementation process is shown in Fig. 4.

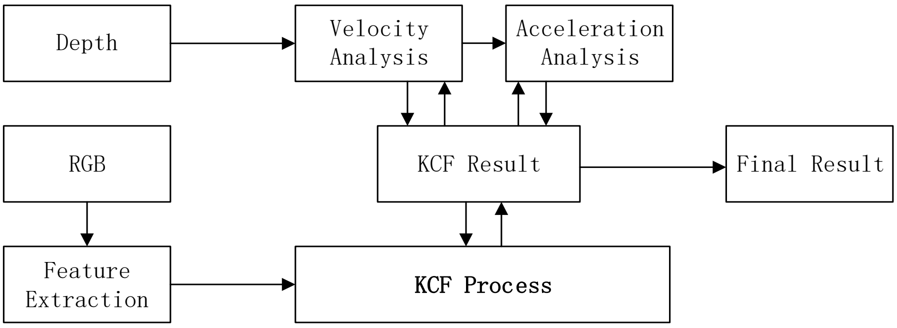

Fig. 4 The fusion of KCF tracker and depth image

Since the rate of change of the target center in the depth image depends on the relative velocity of the quadrotor and the target, this constraint can be introduced to the KCF tracker to suppress the target mismatch due to scale changes and fast motion.

After several experiments，the acceleration in the center of the target in the depth image is shown in Fig. 5. Through the analysis of the experimental data, we can find that when the mismatch occurs, the position coordinates of the target in the depth image is a random distribution in the time domain, so the rate of change is nonlinear in the time domain. However, by analyzing the relative speed of the target and the quadrotor, we can find that the variation of velocity is below a threshold, and the threshold is limited by the velocity matrix $\boldsymbol{V}_t$ of optical flow, so we can set this threshold as the constraint of KCF tracker.

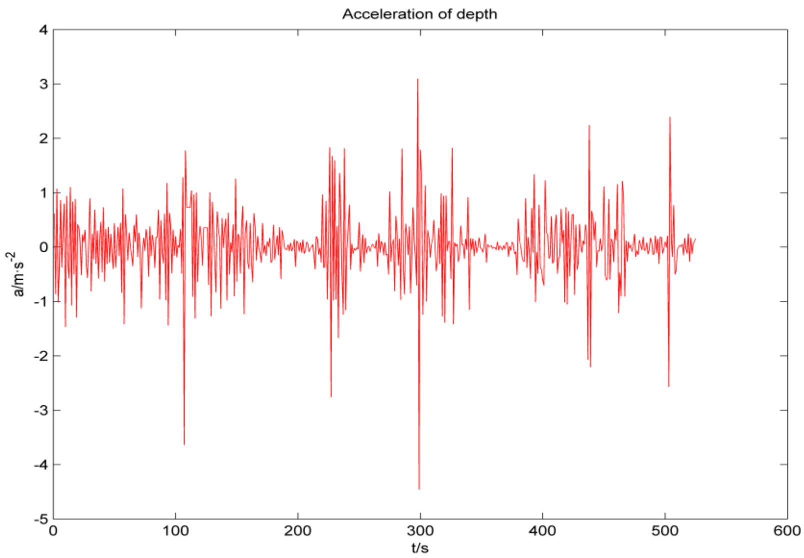

Fig. 5 Acceleration in the center of the target in the depth image

The acceleration of optical flow $\mathbf{a}_t = \boldsymbol{V}_t - \boldsymbol{V}_{t-1}$ .And the tracking result R_t can be defined True or False.

$$R_t = \begin{cases} f''(z) > \mathbf{a}_t, False \\ f''(z) \le \mathbf{a}_t, True \end{cases} \tag{19}$$

If the R_t is False, the system will ignore the tracking result and search the target again, until R_t is true.

Finally, we get the center $m\ (x_p, y_p)$ of the target M under frame coordinates, and z_c under camera coordinates.

As shown in Fig. 6, the left is the depth image and RGB image of previous frame. When the target moves quickly, the accurate tracking of the target is achieved through our algorithm, and the tracking results are shown on the right picture.

Fig. 6 Introduce target tracking of depth images

3.3 Converting the target coordinates to the world coordinate system

Since the camera is running with the drone, a reference coordinate is selected in the environment to describe the camera's position and used to describe the position of any object in the environment. This coordinate system is called the world coordinate system. The following relationship exists between the camera coordinate system and the world coordinate system.

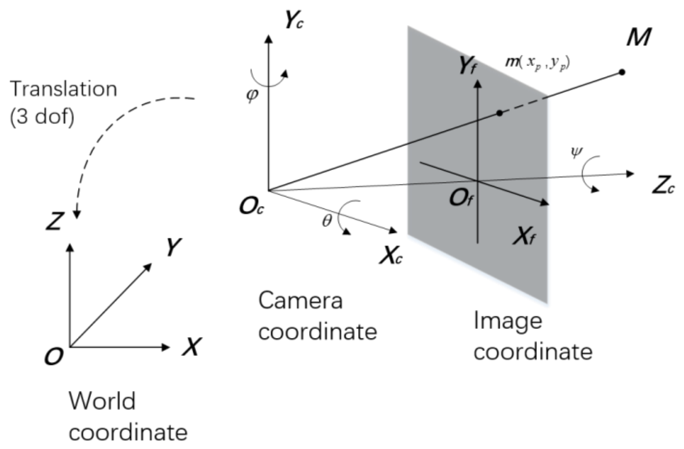

Fig. 7 Coordinate translation for target

Through the processes of visualization described in Fig 7, we can get (X_c, Y_c, Z_c) under the coordinates (x_p, y_p) and the focus f of camera.

$$Z_c \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{20}$$

According to the mathematical relationship in formula, the coordinates of the camera coordinate system can be transferred to the world coordinate system.

Measure the angle (φ, θ, ψ) of the quad rotor through the IMU; we can get the additive effects R of the camera around the *X-Y-Z* three-axis rotation.

$$R_1 = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{21}$$

$$R_2 = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{21}$$

$$R_3 = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{22}$$

$$R = R_1 \times R_2 \times R_3 = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} \tag{23}$$

Finally, the mathematical derivation based on the above formula， we get the (x_t, y_t, z_t) of target under world coordinate.

4. Testing

By analyzing center coordinates obtained of the target area from the image tracking, we can find that when the mismatch occurs, the position coordinates of the target area in the depth image is a random distribution in the time domain, so the rate of change is nonlinear in the time domain. However, analyzing the acceleration of the target and the relative motion of the quadrotor, we can find that the variation of the acceleration is below a threshold. Since this characteristic, we can set this threshold as the constraint of KCF to eliminate the mismatch and get the target center coordinates exactly.

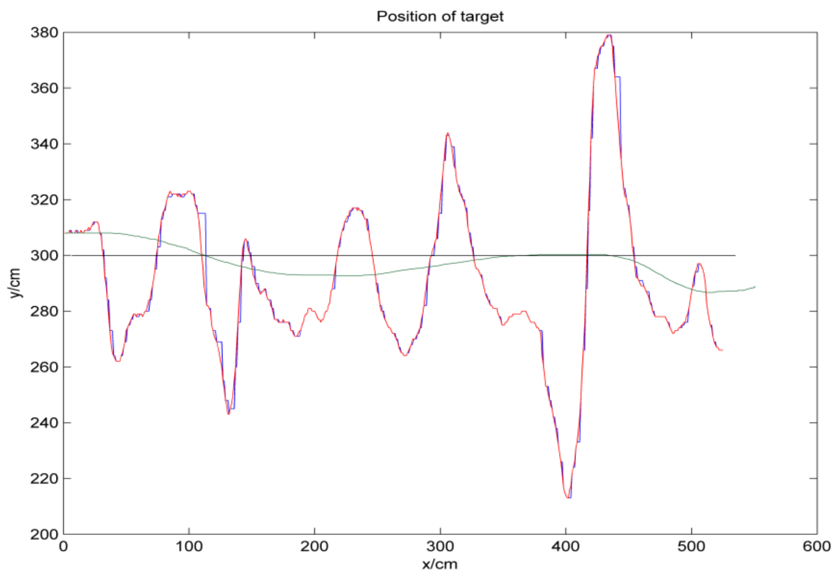

Fig. 8 Determination of the target location

This test on the airborne target tracking results error comparison table:

Table 1

Error comparison table

Approach	**Average error (cm)**	**Maximum error (cm)**
KCF	35	80
KCF + *f* '(*depth*)	33	78
KCF + *f* "(*depth*)	9	17

As shown in the Fig. 8, the black curve is the calibrated target position coordinate, the red curve is the untreated target position coordinate, and the green curve is the modification result of the mismatch coordinates by the acceleration change threshold. The method corrects the mismatch target it improves the accuracy of target tracking and solves the problem that the target is easily lost in the rapid motion of the quadrotor.

5. Conclusion

The proposed method of airborne target positioning and tracking based on multi-information fusion is presented in this paper, which overcomes the deficits of the KCF tracker such as low precision, poor robustness, and error matching. This method realizes the positioning and tracking with the error reduced by 70.41%. The experiments show that it is of high effectiveness and accuracy.

Acknowledgement

This work was supported by the Natural Science Foundation of Shaanxi Province of China (2016GY-007) and the Service Local Special Program of Shaanxi Province Education Department (15JF023). Meanwhile, it was financially supported by Xi'an University of Science and Technology.

R E F E R E N C E S

[1] *Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.:* Accurate scale estimation for robust visual tracking. in: BMVC, PP. 38.1–38.11 2014A

[2] *G. Dong Z Q.* THE measurement of position and qrientation of 6-dof cabin based on the computer vision. Chinese Journal Of Scientific Instrument, (01):65-68+85. DOI:10.3321/J.ISSN:0254-3087.2004.01.015. 2004.

[3] *LI J.* The theory and method of pedestrian traffic video detection in urban traffic system[D]. Beijing Jiaotong University, 2010

[4] *Jean-Yves Bouguet,* Pyramidal Implementation of The Lucas Kanade Feature Tracker Description of the Algorithm, Intel Corporation Microprocessor Research Labs

[5] *Wei W, Dong X Q, Shen Y Y.* Research on 3D model description of human motion in binocular camera system. Microcomputer its Applications, (11):41-44+48. 2011

[6] *Staicu S.* Dynamics of the spherical 3- UPS/S parallel mechanism with prismatic actuators. Multibody System Dynamics, Springer, 2009, 22(2):115-132.

[7] *Staicu S.* Dynamics equations of a mobile robot provided with caster wheel. Nonlinear Dynamics, 2009, 58(1-2):237-248

[8] *PAN G Y.* Optical flow field algorithm and application in video object detection[D], Shanghai Jiao Tong University,2008

[9]. *Kehl W, Tombari F, Ilic S, et al.* Real-Time 3D Model Tracking in Color and Depth on a Single CPU Core[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society,465-473. 2017.

[10]. *Bolme, D., Beveridge, J., Draper, B., Lui, Y.M.:* Visual object tracking using adaptive correlation filters. In: IEEE CVPR, pp. 2544–2550 ,2010.

[11]. *Chen, Z., Hong, Z., Tao, D.:* An experimental survey on correlation filter-based tracking. CoRR abs/1509.05520,http://arxiv.org/abs/1509.05520,2015.

[12]. *Galoogahi, H., Sim, T., Lucey, S.:* Multi-channel correlation filters. In: IEEE ICCV, pp. 3072–3079 ,2013.

[13]. *Henriques, J.F., Caseiro, R., Martins, P., Batista, J.:* High-speed tracking with Kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37(3), 583–596, 2015.

[14]. *Camplani, M., Paiement, A., Mirmehdi, M., Damen, D, Hannuna,S., Tao, L., Burghardt, T.:* Multiple human tracking from RGB-D data: a survey. IET Comput. Vis. (to appear) 2017.

[15]. *Kalal, Z., Mikolajczyk, K., Matas, J.:* Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. 34(7), 1409–1422 ,2012.

[16]. *G. Nebehay and R. Pflugfelder,* "Clustering of Static-Adaptive correspondences for deformable object tracking," in Computer Vision and Pattern Recognition. IEEE, Jun. 2015.

[17]. *Baozhang L, Fengshan H.* A laser distance tracking 3D coordinate vision measuring system[C]// International Conference On Measuring Technology & Mechatronics Automation. IEEE, 83-87. 2011.