Monocular Long-term Target Following on UAVs

Rui Li [*] Minjian Pang[†] Cong Zhao [‡] Guyue Zhou [‡] Lu Fang [†§]

Abstract

In this paper, we investigate the challenging long-term visual tracking problem and its implementation on Unmanned Aerial Vehicles (UAVs). By exploiting the inherent correlation between Frequency tracker And Spatial detector, we propose a novel tracking algorithm, denoted as FAST. As can be theoretically and analytically shown, the superior performance of FAST originates from: 1) robustness – by transforming from frequency tracker to spatial detector, FAST owns comprehensive detector to cover consequential temporal variance/invariance information that inherently retained in tracker; 2) efficiency – the coarse-to-fine redetection scheme avoids the training of extra classifier and exhaustive search of location and scale. Experiments testified on tracking benchmarks demonstrate the impressive performance of FAST. In particular, we successfully implement FAST on quadrotor platform to tackle with indoor and outdoor practical scenarios, achieving real-time, automatic, smooth, and long-term target following on UAVs.

1. Introduction

With their extremely high flexibility, portable size and fast speed, UAVs have emerged as a rising star among mobile robots in recent years. Endowing the UAVs with intelligent vision based algorithms is in urgent demand, and one of the most fundamental intelligent features apparently lies in automatic target following via a long-term visual tracking method so as to push consequential applications of UAVs far beyond amusement to surveillance [5], augmented reality [10], behavior modeling [23] *etc*.

Contrary to the GPS-based target following on UAVs, which requires the target to wear an extra GPS-equipped device for communication [26] and is incapable in GPS-denied environments (*e.g*., indoors, urban areas), we propose a more universal and flexible vision based tracking method, which is successfully implemented on a DJI drone platform (Fig. 1) to perform on-board long-term target following for both indoor and outdoor practical scenarios.

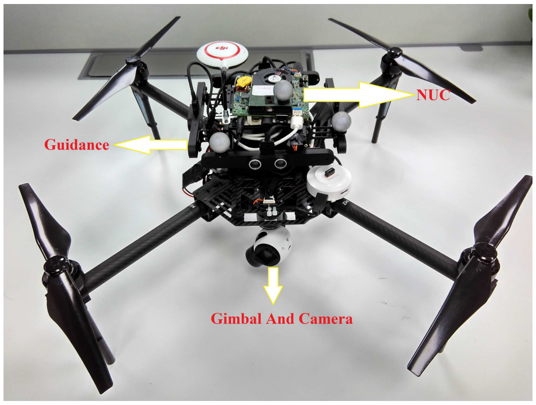


Figure 1. Our on-board platform: DJI *Matrice 100* with Intel NUC (i5-4520u), DJI *Guidance*, DJI *Zenmuse X3* gimbal and monocular camera.

Given the fact that long term tracking remains a challenging problem due to complex factors (*e.g*., deformation, occlusion, *etc*.) in real application scenarios, a recovery mechanism should be integrated into the framework to restart tracking when severe failure occurs. However, existing works pay limited attention to redetection by roughly training an extra classifier for detection together with performing an exhaustive search of location and scale, thereby ignoring the important temporal context and being far from efficient [17, 20, 30, 37]. Thus, in this paper we investigate the questions: 1) *Is the conventional way that trains an extra classifier for redetection really essential, and* 2) *if tracking by detection is well recognized, can detection by tracking work?* Interestingly, as will be theoretically and analytically shown, an online training tracking model can be regarded as a well-trained classifier for redetection, implying the inherent connection between tracker and detector. Thus we can answer 'NO' and 'YES' to these two questions.

Standing on our revelation, we present a novel long-term visual tracking scheme via Frequency And Spatial Transformation (denoted as *FAST*) that owns adaptive model updating and a failure recovery mechanism. Specifically, *FAST* adopts a frequency domain correlation filter as the base

[*] alr@mail.ustc.edu.cn, University of Science and Technology of China, Hefei, Anhui, China

[†] {mpangaa, eefang}@ust.hk, Hong Kong University of Science and Technology, Hong Kong, China

[‡] {cong.zhao, guyue.zhou}@dji.com, Dajiang Innovations Technology Co., Ltd., Shenzhen, China

[§] Corresponding author

tracker, whose online training model is transformed to the spatial domain as the detecting model. To resolve the problem that redetection schemes often suffer exhaustive search of location and scale, a coarse-to-fine two-stage redetection scheme is also proposed.

In summary, the technical contributions of this paper are: 1) we explore the inherent connection between the frequency domain tracker and spatial domain detector, showing that the tracking model can be severed as a well-trained classifier for redetection; 2)we also propose a generic coarse-to-find redetection scheme that contains *a generic object proposal for coarse selection* and *discriminative detector for find detection*, which effectively avoids the exhaustive search of location and scale; 3) we also propose an efficient framework for the practical target following problems on UAVs in GPS-denied environments, which integrates target 3D localization, self-localization, flight control, *etc*.

2. Related Work

In this section, we review the state-of-the-art target following schemes on UAVs and visual tracking algorithms.

2.1. Target Following on UAVs

Tracking and detection play important roles in numerous applications in robotics. Many navigation and following problems require accurate location estimation as a control signal for posture adjustment (*e.g*., aerial refueling [36], tracking [26, 28, 40], navigation [31], pedestrian tracking [29]). Serving as the most widely used tool for object localization, the GPS-based method requires GPS-equipped devices to receive the locating signal. Despite the requirement of an extra device, the performance of GPS-based method tends to be attenuated or incapable in GPS-denied environment(*e.g*., indoors and in urban areas).

Computer vision technologies, which can be treated as a powerful sensor to perceive the world around us, bring more intelligence to robots. [36] trained a linear svm for detecting and tracking a drogue object of aerial refueling via simple LBP feature. [29] adopts Aggregated Channel Features (ACF) for detecting pedestrians and utilizes particle filter for avoiding frequent detection. However, these methods are designed for a specific object category and require offline training, which go against generic object following. [13] adopts correlation filter for generic object tracking, but a simple short-term tracker can not effectively handle complex environments. [24] and [25] adopt openTLD [17] for target tracking and detection on a UAV platform, however, the TLD approach can not achieve comparable performance to state-of-the-art tracking algorithms. In contrast to previous schemes, we propose a novel long-term visual tracking scheme that achieves a good balance of effectiveness and efficiency for target following on UAVs.

2.2. Visual Tracking Algorithm

Recently, many novel visual tracking approaches have been proposed and achieve significant improvements; however, existing algorithms are generally deficient in solving the real-time long-term target following problems on UAVs.

Early approaches of template based tracking work by finding an optimal patch to describe appearance via selectively updating template [21], abd multiple key frames [27]. Although template based methods are robust to appearance variation [6], they usually do not fully utilize previous object information, and lose most of the object structure in previous features. Sparse representation is another approach to find sparse appearance patterns [19, 32, 39]. However, these methods are computationally intensive and do not fully utilize correlation among frames.

Tracking-by-detection methods adopt a detection method to discriminate the object and background from sequential images, where early approaches exploited time-invariant information of tracking the target via an online training model, *e.g*., SVM [2, 4, 14] and boosting [3, 11, 12]. These methods directly adopt detection a framework to solve the tracking problem, and generally fail to utilize the temporal context.

Correlation filter [16] emerges as one of the most successful tracking frameworks due to its efficiency, accuracy and simplicity. Many state-of-art trackers are built upon correlation filter, *e.g*., [7–9]. The exceptional performance gain lies in that it fully utilizes current frame information (by circular shift) with a dense search. However, its online update scheme tends to tie to short-term object appearance, and forget the former information gradually. Thus, it is extremely powerful in short-term tracking, but is relatively vulnerable to appearance variation.

For improving the robustness of tracking, [17] propose TLD that decomposes long-term tracking into three main components: tracking, learning and detection. To deal with long-term out-of-view, occlusion, [35] address the problem as detecting occlusion and variation of view-point by combining occlusion and motion reasoning. [37] online trains multiple experts, and selects the best expert to correct the current tracking result, which is effective to protect tracking model from drifting. [20] propose a long-term tracking method by using PSR to estimate failure, and train an extra classifier for redetection.

In contrast to previous methods, we exploit the model relationship in frequency and spatial domain, and propose a novel long-term tracking scheme via utilizing the current tracking model as a well-trained detector, which is free from training extra classifier. Our method is further implemented on a DJI Quadrotor platform, achieving excellent performance of target following on UAVs.

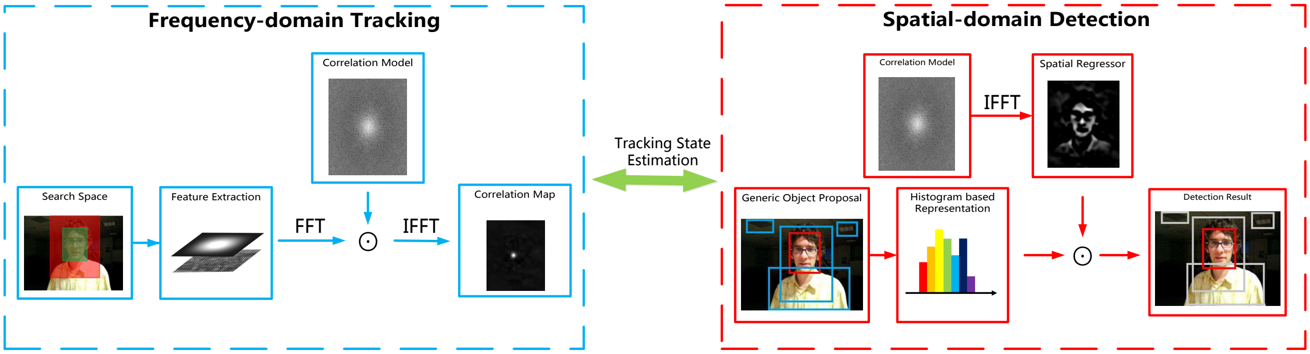


Figure 2. Overview of our proposed long-term tracking algorithm via frequency and spatial transformation: *FAST*.

3. Proposed Frequency and Spatial Transformation for Tracking (FAST)

As shown in Fig. 2, *FAST* can be divided into two modules: the frequency domain tracking module serves to estimate the translation and scale of target from previous observations and the spatial domain redetection module aims at estimating tracking quality (*e.g*., success, failure) and searching for the reappearing target when severe tracking failure occurs.

3.1. Frequency Domain Discriminative Model

Given the graceful mathematical expression and efficient implementation of correlation filter [16], we take correlation filter as our discriminative model, which trains a tracking model in frequency domain without the requirement of extracting the positive and negative examples, and updates via simple incremental learning. Specifically, we consider all previous frames to train our discriminative model (denoted as $\mathbf{w}$) via ridge regression as follows*,

$$\min_{\mathbf{w}} \sum_{k=1}^{p} \alpha_k \sum_{i,j} |\langle \mathbf{x}_{ij}^{k}, \mathbf{w} \rangle - y_{ij}|^2 + \lambda \|\mathbf{w}\|^2, \qquad (1)$$

where k denotes the frame index, p is the number of total frames and α_k is the weight for k-th frame. $\mathbf{x}_{ij}^{k}$ is cyclic shift of the feature map (of size $W \times H$) for $(i,j) \in \{0, \cdots, W\} \times \{0, \cdots, H\}$ on k-th frame and y_{ij} is the Gaussian-shaped regression target. $\langle \cdot \rangle$ is dot product operation and $\lambda > 0$ is the regulation parameter. Eqn. (1) can be solved and accelerated in frequency domain, so that

$$\mathbf{W} = \frac{\sum_{k=1}^{p} \alpha_k \overline{\mathbf{X}}_k \odot \mathbf{Y}_k}{\sum_{k=1}^{p} \alpha_k (\overline{\mathbf{X}}_k \odot \mathbf{X}_k + \lambda)}, \qquad (2)$$

where $\mathbf{W} = \mathcal{F}(\mathbf{w})$, $\mathcal{F}$ denotes the Fast Fourier Transform (FFT) operation, $\odot$ is the Hadamard product, $\bar{\cdot}$ is the conjugate operation, and the division is element-wise. Discriminative model $\mathbf{W}$ can be optimally updated as follows,

$$\begin{aligned} \mathbf{N}_p &= (1-\eta)\mathbf{N}_{p-1} + \eta \overline{\mathbf{X}}_p \odot \mathbf{Y}_p, \\ \mathbf{D}_p &= (1-\eta)\mathbf{D}_{p-1} + \eta (\overline{\mathbf{X}}_p \odot \mathbf{X}_p + \lambda), \end{aligned} \qquad (3)$$

where the discriminative model is represented by $\mathbf{W} = \frac{\mathbf{N}}{\mathbf{D}}$, $\eta \in [0,1]$ is the learning rate, and $\alpha_k = \eta(1-\eta)^{p-k}$.

$\mathbf{R}$ denotes the correlation map of feature patch $\mathbf{Z}$ and tracking model $\mathbf{W}$ in frequency domain, given by

$$\mathbf{R} = \mathbf{W} \odot \mathbf{Z}. \qquad (4)$$

Taking inverse FFT of $\mathbf{R}$, we have $\mathbf{r} = \mathcal{F}^{-1}(\mathbf{R})$ – the correlation map in spatial domain (regression result for each cyclic shift of patch), whose peak exactly indicates the most confident target position for estimating translation.

For handling scale variation, similar to [7], we initialize scale filter by resizing the initial tracking patch to a fixed size, where scale filter can be trained via Eqn. (2). Then the scale pyramids are built around the tracking target after translation estimation, and the scale with the maximum correlation is selected as the current tracking scale. The update scheme of scale filter is similar to Eqn. (3).

3.2. Spatial Domain Coarse-to-fine Redetection

As reviewed in the related work, regardless of the outstanding performance, a robust long-term tracking scheme is expected to have its own recovery mechanism, *i.e*., estimating tracking state and redetection if needed. Serving for failure detection so as to enable redetection, we estimate tracking state by Peak-to-Sidelobe Ratio (PSR) $\tau = \frac{r^* - \mu}{\sigma}$, where r^* is the maximal value on the correlation map $\mathbf{r}$, and we define the sidelobe as the $N \times N$ image patch ($N = 10$). μ and σ are the mean and standard deviation of the sidelobe respectively. Given τ, we simply define three tracking states

*For ease of presentation, we define $\mathbf{x}$ as a single channel feature, which can be extended to multi-channel features easily. The lowercase matrix corresponds to the spatial domain, and the capital-letter matrix corresponds to the frequency domain.

based on two predefined thresholds τ_l and τ_h for failure detection: success, borderline, and fail. If state is success, current model is updated by averaging new information; and if the state is borderline, tracking model will not update since new information may introduce a drift problem; and if the state is failure, redetection start. Contrary to most tracking methods that pay limited attention to redetection by roughly training an extra classifier as detector followed by exhaustive search of location and scale, which apparently neglects the important temporal context and is far from efficient, we propose a coarse-to-fine two stage redetection scheme to search for reappearing object as follows.

Object Proposal for Coarse Selection As object may reappear in various locations and scales after tracking failure, an exhaustive search by sliding window is incompetent in predicting the significant scale variation efficiently. Generic object proposal [41], however, impliedly deals with the scales problem by providing a small set of object candidates with numerous aspect ratios and sizes without the bias of categories. Simply adopting generic object proposal leads to redundancy that most bounding boxes are less likely to bound the tracking object due to the lack of target-dependant information, *e.g.*, color distribution, aspect ratio and size. To handle scale variation while further reducing computational burden, target-related features are utilized for bias selection given the fact that reappearing object may vary in location and scale but rarely in aspect ratio or color distribution.

We generate initial object proposals in concert with aspect ratio of initial tracking target, and calculate objectness score via the method in [41]. For each initial bounding box, we adjust the target rectangle until maximal objectness score is obtained. In this refinement stage, object proposal will be adjusted to local optimal position to bound the nature object, as well as consider the aspect ratio of possible candidates. We rank object proposals via considering the color similarity and objectness score as follows:

$$S_c = S_{color} * S_{obj}, \tag{5}$$

where S_{color} is calculated by the Euclidean distance of histogram between initial frame and object proposal patch.

While coarse selection via generic object proposal succeeds constraining the candidates to a certain amount efficiently, it is essential to further design fine selection to achieve the prime target candidate. In our method, only top 10 object proposals are selected for the next stage detection.

Discriminative Detector for Fine Selection An intuitive solution to detect the prime target candidate is training an extra classifier via extracting positive and negative samples from previous frames [17, 20]. However, such a scheme actually preserves invariant information of target and does not explicitly carry temporal correlation, since it treats each frame equally for training the classifier [22]. We further find that 1) *the tracking model inherently contains temporal context* for all of the previous frames and weighting by time order, leading to the better description of the current object appearance, 2) the analysis of redetection reveals that *the tracking model can serve as a discriminative detector for redetection*. We will elaborate these two claims in following contents.

1) *Tracking model inherently contains temporal context*. Recall that in Eqn. (3) the discriminative model $\mathbf{W}$ can be optimally updated as

$$\begin{aligned} \mathbf{N}_p &= \sum_{k=1}^{p} \eta(1-\eta)^{p-k}\overline{\mathbf{X}}_k \odot \mathbf{Y}_k, \\ \mathbf{D}_p &= \sum_{k=1}^{p} \eta(1-\eta)^{p-k}(\overline{\mathbf{X}}_k \odot \mathbf{X}_k + \lambda), \end{aligned} \tag{6}$$

where $\mathbf{X}_k$ and $\mathbf{Y}_k$ are k-th frame information. As we expected, the weighting term $\eta(1-\eta)^{p-k}$ decreases as time goes by for weighting the temporal information, implying that correlation filter model inherently owns advantage to reflect the temporal variation of the tracking object as well as invariant information.

2) *Tracking model can serve as a discriminative detector for redetection.* Recall that redetection can be modeled as the correlation calculation between detection model $\mathbf{w}$ and candidate patch $\mathbf{z}$ in spatial domain utilizing ridge regression model from Eqn. (1). It is known that Fourier transform maps correlation in spatial domain to multiplication in frequency domain, *i.e.*,

$$\mathbf{R} = \overline{\mathbf{W}} \odot \mathbf{Z}, \tag{7}$$

where $\bar{\cdot}$ denotes conjugate operation. Eqn. (7) is the formal definition of correlation, which exactly resembles Eqn. (4) but $\mathbf{W}$ requires conjugate operation. Correlation filter can be updated optimally via Eqn. (3), we can directly adopt the tracking model in frequency domain to calculate regression result in spatial domain as follows:

$$\hat{\mathbf{w}} = \mathcal{F}^{-1}(\overline{\mathbf{W}}). \tag{8}$$

We evaluate the regression result between the spatial model and candidates $\mathcal{C}$ – the candidate set after coarse selection by object proposal (where all the selected candidates are resized to fit the size of translation model) via

$$\tilde{r} = \langle \hat{\mathbf{w}}, c \rangle, \ \ c \in \mathcal{C}. \tag{9}$$

Simple dot-product between the candidate and translation model in spatial domain can approximate the correlation of specific cyclic shift. Therefore, the optimal candidate (denoted as c^*) of reappearing target is determined via

$$c^* = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \langle \hat{\mathbf{w}}, c \rangle. \tag{10}$$

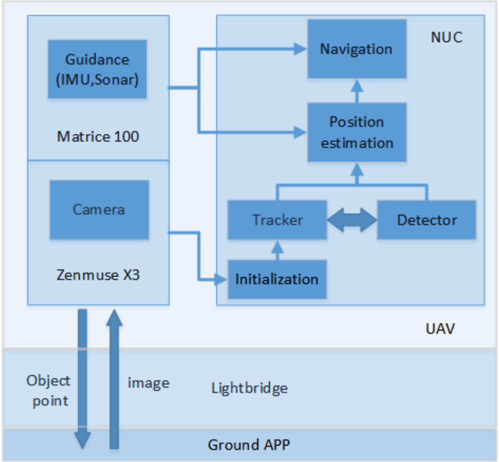


Figure 3. The overall architecture of our method on DJI *Matrice 100* platform. Guidance equipped with IMU and sonar provides the altitude and speed of UAVs, camera acquires image data, NUC is computational unit for processing image and return control signal, Lightbride communicates between aerial and ground station.

To ensure more robustness regarding the false alarm, we also estimate the confidence of detection result via the PSR measurement, and compute the correlation map in frequency domain via Eqn. (4) with new object position and scale. The new position and scale are accepted under the case that the PSR of new object position is larger than the predefined threshold τ_h. In our simulation, the object proposal [41] reports a recall of $96\%\backslash87\%$ at threshold of $0.5\backslash0.7$ for 1000 proposals on popular object detection benchmark VOC2007, leading to high possibility of discovering reappeared object when it adopts object proposal for redetection.

Our model updates in the frequency domain, and is transformed to the spatial domain by inverse FFT. Dot-product operation is used for calculating the correlation of spatial model and candidate. Candidate with maximal correlation is accepted as the new tracking target if it satisfies the failure detection condition. Regarding the computational complexity, directly calculating correlation for all candidates by Eqn. 4 requires $|\mathcal{C}|$ FFTs, where $|\mathcal{C}|$ is the element number of $\mathcal{C}$. While the spatial detector merely requires one inverse FFT. More comparisons of efficiency are reported in the experiment section.

4. System Architecture

Our algorithm is implemented on the DJI *Matrice 100* quadrotor platform (Fig. 1), where the video sequence is given by a monocular camera, and all the on-board computations are processed on Intel NUC (i5-4520u). The *Matrice 100* includes DJI *Lightbridge* – the high definition image transmission system to enable initialization on the ground. The overall system architecture is shown in Fig. 3, where all the algorithms (Fig. 4) can run on NUC in real time.

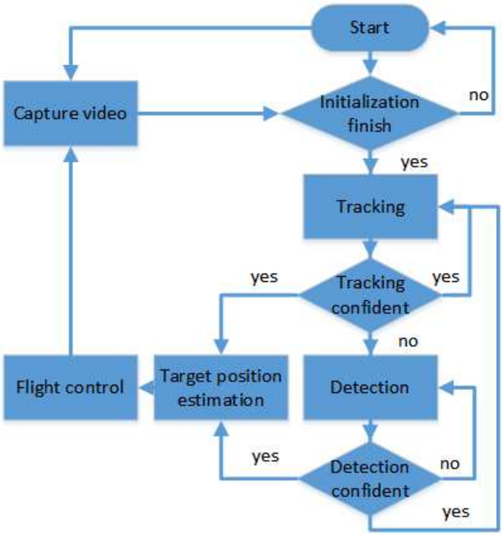


Figure 4. The software pipeline of long-term visual tracking scheme and flight control.

Initialization Recall that the images captured by the monocular camera of UAV can be transmitted to ground app via *Lightbridge* expeditiously, we can then simply select a point on the target that will be sent back to UAV immediately, with which, our on-board algorithm then identifies the desired object shaped by rectangular bounding box. Specifically, we use Edgebox [41] to return a set of object proposals – well-organized bounding boxes that likely contain objects. The proposals that contain a user-clicked point are further averaged to assist initialization efficiently. Thus, it is more efficient than manually drawing a rectangle for desired object, and is more user-friendly as most users have little sense on initializing a 'good' bounding box.

5. Target Position Estimation and Navigation

Given the tracking bounding box (2D), the distance estimation method proposed in this section serves to obtain the 3D position (*i.e.*, the relative orientation and distance between the UAV and target) of target continuously and stably. We propose novel method to estimate distance between target and UAVs under three assumptions: 1)object height is relatively stable, *e.g.*, rigid objects; 2) the roll angle for camera is 0 (gimbal system has such guarantee); 3)UAV height is available at initial stage by IMU.

Target Position Estimation Let us start the discussion for the initialization stage as illustrated in Fig. 5; we denote $\overrightarrow{T}$, $\overrightarrow{B}$ as the top and bottom constrains of bounding box in North-East-Down coordinate, *i.e.*, $\overrightarrow{T}$ and $\overrightarrow{B}$ are the highest/lowest points of target respectively, and can be rep-

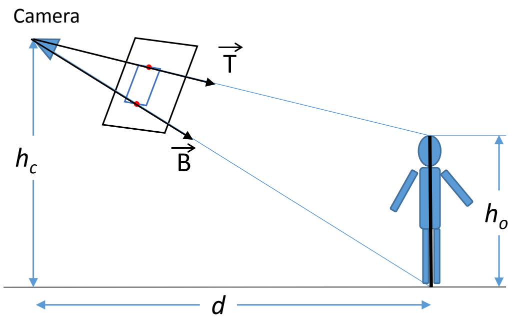


Figure 5. Illustration of target position estimation at initialization stage.

resented by

$$\vec{T} = \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \sim \mathbf{RK}^{-1} \begin{pmatrix} u_t \\ v_t \\ 1 \end{pmatrix}, \qquad (11)$$

$$\vec{B} = \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} \sim \mathbf{RK}^{-1} \begin{pmatrix} u_b \\ v_b \\ 1 \end{pmatrix}. \qquad (12)$$

where $\mathbf{K}$ is the intrinsic matrix of the camera, $\mathbf{R}$ is the camera rotation, (u_t, v_t) and (u_b, v_b) represent the red dots in Fig. 5 in the 2D image coordinate, which are the intersections of $\vec{T}$ and $\vec{B}$ on 2D image, giving the top and bottom constrains of bounding box in 2D image, respectively.

Then, the distance between the target and UAV is given by $d_0 = \frac{P_b}{z_b} \times h_c$, the height of target is $h_o = h_c - \frac{z_t}{P_t} \times d_0$, where h_c is the height of the UAV retrieved from IMU, and h_o is the height of object. $P_b = \sqrt{x_b^2 + y_b^2}$ and $P_t = \sqrt{x_t^2 + y_t^2}$ are the projection length of $\vec{B}$ and $\vec{T}$ on the ground, respectively. Given the distance and height at initialization stage, for every upcoming frame that has a bounding box, the distance between the target and UAV is

$$d_t = h_o / || \frac{\vec{T}}{P_t} - \frac{\vec{B}}{P_b} || = h_o \times \frac{P_t}{||\vec{T} - \vec{B}||} \quad (P_t \approx P_b), \qquad (13)$$

Note that after initialization, our system does not require the tracking target to be on the ground or knowing the height of UAV. This is especially valuable to the cases where target object climbs up and moves down during tracking as well as the height of UAV, since IMU may not be reliable when the UAV flies over grasses or higher than 5 meters.

Flight Control With the estimated target position that smoothed by Kalman filter, the corresponding flight control in our system controls the UAV in at desired distance. Specifically, the relative velocity of the target compared to the drone is calculated, with which, the relative velocity of the target compared to the ground is obtained by knowing the velocity of the drone retrieved from IMU of the UAV. After collecting the position and velocity of the target, a PID controller is utilized to assure a safe distance between the drone and target. In practice, instead of controlling the drone both horizontally and vertically, our system controls the gimbal's yaw angle first and then controls the drone's yaw to follow the gimbal, i.e., to align the drone with the gimbal's yaw orientation. The reason lies in the fact that it is more flexible for a drone to turn around than move left or right due to the inertia. Thus, for most cases, the target is in the center of the image, and the drone will smoothly slow down and hover once the target is lost, and will react once redetection is performed to identify the target again.

6. Experimental Results and Discussions

In this section, we first present extensive evaluations of the proposed tracking algorithm *FAST* [†] and state-of-the-art visual tracking methods. Then, we evaluate the implementation of *FAST* on the DJI UAV platform for both indoor and outdoor scenarios from two aspects: target position estimation and flight control. **The video demonstration of our algorithm and implementation are provided as supplementary in** `https://youtu.be/akBddFrw6Nk`.

Our work is implemented by C for UAV system with an i5-4520u 1.6 GHz CPU and for simulation on PC with a i3-2100 3.1 GHz CPU.

6.1. Evaluation of *FAST* on Tracking Benchmark

We mainly report our results on the popular benchmark proposed by [34], which contains 50 sequences and is annotated with 11 attributes that indicate challenging types. Several tracking algorithms treated as the state-of-the-art methods include DSST [7], CSK [15], KCF [16], STC [38], Struck [14], TLD [17], MEEM [37] and RPT [18]. All the concerned methods are assessed by widely-used metrics [33]: *Precision plot*, which computes the percentage of frames in which the estimated locations are within a given threshold of the ground truth positions; *Success plot*, which computes the percentage of frames in which the overlap between the estimated and ground truth bounding box is within given thresholds. In particular, Distance Precision (DP) at 20 pixels, and more challenging, at 10 pixels, and Overlap Success (OS) at an overlap threshold 0.5 and 0.7 are discussed. Regarding *Robustness* evaluation, the metrics includes: One Pass Evaluation (OPE) that starts tracking at first frame; Temporal Robustness Evaluation (TRE) that starts tracking at different frames; and Spatial Robustness Evaluation (SRE) that starts tracking at different bounding boxes with sight shifting or scaling, respectively.

[†] In our experiments, regularization parameter $\lambda = 0.01$ and learning rate $\eta = 0.025$. The search of the tracking window size is two times that of the target size. The number of scale search spaces is 33, and the scale factor is 1.02. The thresholds for state estimation are $\tau_l = 8$ and $\tau_h = 12$.

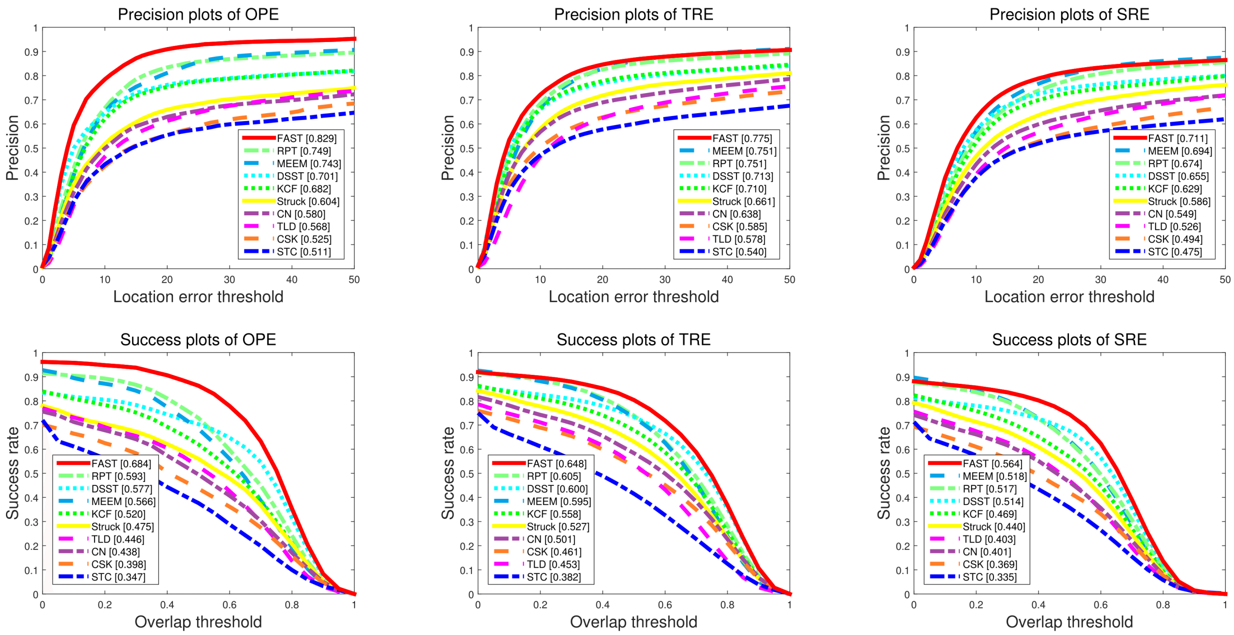


Figure 6. **The evaluation of the precision plot and success plot with OPE, TRE and SRE measure for all 50 sequences.** The concerned stat-of-the-art methods include CSK [15], KCF [16], STC [38], Struck [14], TLD [17], MEEM [37], CN [9] and RPT [18]

In Table 1, we present the evaluations of DP, OS and FPS. RPT [18] is considered to be state-of-the-art, and achieves outstanding performance in DP=0.834/0.666 under the thresholds 20px/10px, and OS=0.729/0.449 under the thresholds 0.5/0.7. Our *FAST* shows a comparably superior performance to the state-of-the-art, in terms of the best DP of 0.909/0.785 under the distance thresholds of 20px/10px and the best OS of 0.863/0.632 under the overlap threshold 0.5/0.7, respectively.

Algorithm	DP (20px)	DP (10px)	OS (0.5)	OS (0.7)	FPS
FAST	**0.909**	**0.785**	**0.863**	**0.632**	100.8
DSST [7]	0.762	0.645	0.704	0.546	37.4
CSK [15]	0.554	0.424	0.436	0.275	**315.8**
KCF [16]	0.753	0.621	0.620	0.388	161.2
STC [38]	0.554	0.433	0.377	0.199	72.2
Struck [14]	0.659	0.517	0.560	0.367	13.7
TLD [17]	0.613	0.464	0.529	0.298	29.9
CN [9]	0.629	0.499	0.495	0.302	109.2
RPT [18]	0.834	0.666	0.729	0.449	4.0
MEEM [37]	0.811	0.649	0.684	0.386	15.3

Table 1. Comparison with state-of-the-art in terms of the Distance Precision (DP), Overlap Success (OS), and Frame Per Seconds (F-PS). The best performance is highlighted in bold. We calculate the average speed for FAST including tracking and detection; tracking achieves approximately 100 fps and detection achieves approximately 120 fps for the C implementation.

In Fig. 6, we present the evaluation of OPE, TRE and SRE in the precision plot and the success plot. The tracking methods are ranked based on the Area Under the Curve (AUC). It can be shown that *FAST* (red curve) outperforms others in the precision plot of OPE, TRE, SRE and the success plot of OPE, TRE and SRE. In particular, Fig. 7 presents the evaluation of four attributes that reflect challenges in different aspects: deformation, occlusion, out-of-view and out-of-plane, where *FAST* achieves competitive performance in both the precision plot and success plot.

In summary, *FAST* achieves stable and accurate translation/scale estimations that are robust to challenging factors. It is also robust to the challenge cases in short-term tracking, *e.g.*, partial occlusion, illumination variation and deformation, due to the feature of estimating current tracking state for adaptive model updating. *FAST* works well in the more challenging cases of long-term tracking, *e.g.*, out-of-view, occlusion, deformation. Since our coarse-to-fine redetection module generates a detector by transforming the frequency tracking model to the spatial domain, which largely releases the burden of training extra classifier, and the temporal information that is inherently retained in tracker further improves redetection accuracy.

6.2. Implementation of *FAST* on UAV Platform

We further assess the performance of *FAST* on a UAV platform for both indoor and outdoor scenarios.

To objectively evaluate the accuracy of on-board target position estimation, we conduct an indoor experiment by fixing the target while controlling the UAV manually to get plenty of distance between the target and the UAV. The es-

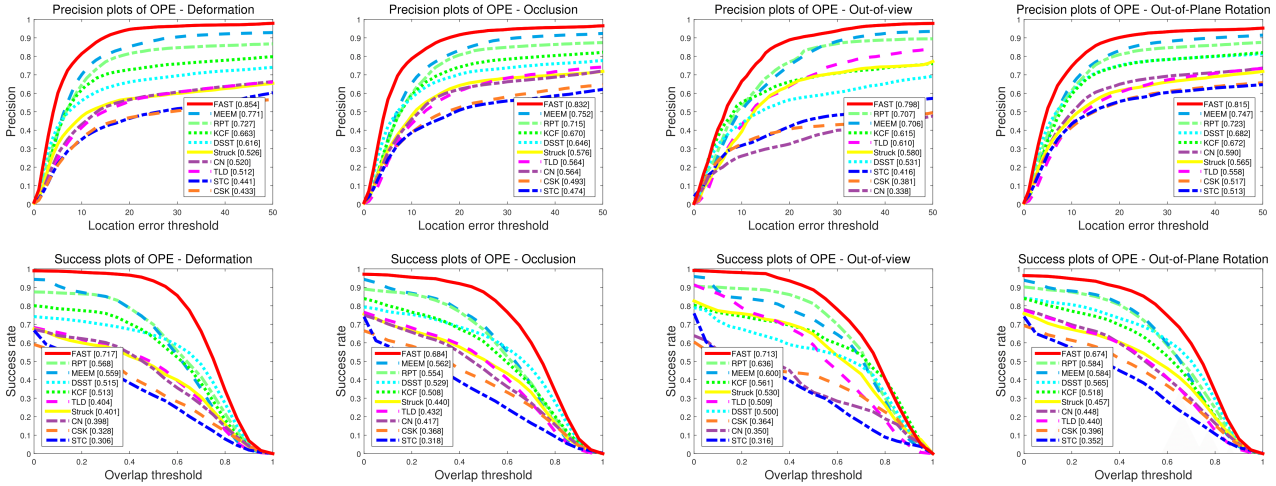
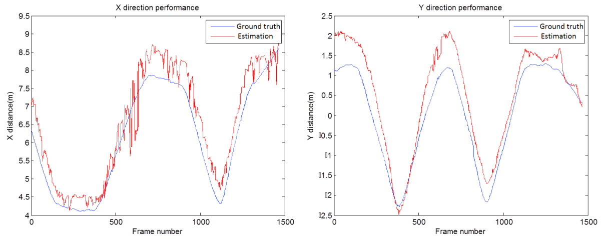


Figure 7. **The evaluation of the precision plot and the success plot for all 50 sequences on selected attributes.** The selected attributes include deformation, occlusion, out-of-view and out-of-plane rotation.



(a) X (forward - backward)

(b) Y (left - right)

Figure 8. Outdoor experiment for comparing tracking accuracy; the red curve is the *FAST* result, and the blue curve is Vicon result.

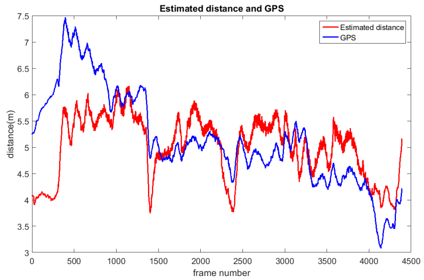


Figure 9. Outdoor experiments for comparing tracking accuracy; the red curve is the *FAST* result, and the blue curve is GPS result.

timated target location in the X (forward-backward) and Y (left-right) directions using our method is shown in Fig. 8-red curve. Note that we directly show the raw data without smoothing filter, thus the red curve appears noisy due to the inherent noise of the UAV hovering and the flight control. With the markers in both the target and the UAV, the Vicon motion tracking system [1] provides an accurate measurement, which serves as ground truth data, as shown by the blue curve. In general, the error of the target position estimation is smaller than $0.5m$, enabling the UAV to act sensitively. One may notice that the estimation in the Y direction tends to be less noisy since we maintain the bounding box of the target in the center of image, even under different scales. Y is still stable while X has to be adjusted frequently.

Our implementation is further validated in the outdoor environment under the case that the UAV automatically operates the flight control without human interference, and the desired distance between the UAV and the target is set to be 5 meters. The performance of our system (red curve) is compared with the GPS data (blue curve) in Fig. 9, where the Y axis denotes the straight-line distance in meters, and the X axis denotes frame numbers. It can be shown that our system is capable of maintaining desired distance between the UAV and the target stably. The GPS varies significantly during the starting period, as it takes time to achieve accurate localization at the beginning.

7. Conclusion

In this paper, we propose a novel long-term visual tracking algorithm *FAST* and implement it on a DJI UAV platform. *FAST* estimates the translation and scale via correlation filter theory and obtains a well-trained object classifier (serving for redetection) via a transforming tracking model from the frequency domain to the spatial domain. Extensive experiments show that *FAST* achieves competitive performance in terms of widely-used evaluation metrics, as well as robust, smooth, long-term target following on UAVs.

Acknowledgements This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 61303151, in part by the GRF 16211615.

References

[1] Motion capture systems from vicon. http://www.vicon.com/.

[2] S. Avidan. Support vector tracking. *PAMI*, 2004.

[3] S. Avidan. Ensemble tracking. *PAMI*, 2007.

[4] Y. Bai and M. Tang. Robust tracking via weakly supervised ranking SVM. In *CVPR*, 2012.

[5] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 2003.

[7] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[8] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[9] M. Danelljan, F. S. Khan, M. Felsberg, and V. D. W. Joost. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[10] V. Ferrari, T. Tuytelaars, and L. Van Gool. Real-time affine region tracking and coplanar grouping. In *CVPR*, 2001.

[11] H. Grabner, H. Bischof, and M. Grabner. Real-time tracking via on-line boosting. In *BMVC*, 2006.

[12] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008.

[13] K. Haag, S. Dotenco, and F. Gallwitz. Correlation filter based visual trackers for person pursuit using a low-cost quadrotor. In *International Conference on Innovations for Community Services (I4CS)*, 2015.

[14] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[15] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 2015.

[17] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 2011.

[18] Y. Li, J. Zhu, and S. C. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *CVPR*, 2015.

[19] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, 2011.

[20] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *CVPR*, 2015.

[21] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 2004.

[22] M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[23] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.

[24] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli. Vision based gps-denied object tracking and following for unmanned aerial vehicles. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013.

[25] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy. Computer vision based general object following for gps-denied multirotor unmanned vehicles. In *American Control Conference (ACC)*, 2014.

[26] S. A. P. Quintero, M. Ludkovski, and J. P. Hespanha. Stochastic optimal coordination of small uavs for target tracking using regression-based dynamic programming. *Journal of Intelligent and Robotic Systems*, 2016.

[27] A. Rahimi, L. P. Morency, and T. Darrell. Reducing drift in differential tracking. *CVIU*, 2008.

[28] P. Sadeghi-Tehran, C. Clarke, and P. Angelov. A real-time approach for autonomous detection and tracking of moving objects from UAV. In *IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*, 2014.

[29] F. D. Smedt, D. Hulens, and T. Goedemé. On-board real-time tracking of pedestrians on a UAV. In *CVPRW*, 2015.

[30] J. S. Supancic III and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013.

[31] Y. Watanabe, P. Fabiani, and G. L. Besnerais. Simultaneous visual target tracking and navigation in a gps-denied environment. In *IEEE International Conference on Advanced Robotics (ICAR)*, 2009.

[32] Z. Wei, L. Huchuan, and Y. Ming-Hsuan. Robust object tracking via sparse collaborative appearance model. *TIP*, 2014.

[33] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *PAMI*, 2015.

[34] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[35] H. Yang, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In *ECCV*, 2014.

[36] Y. Yin, X. Wang, D. Xu, F. Liu, Y. Wang, and W. Wu. Robust visual detection-learning-tracking framework for autonomous aerial refueling of uavs. *IEEE Trans. Instrumentation and Measurement*, 2016.

[37] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[38] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *ECCV*, 2014.

[39] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang. Structural sparse tracking. In *CVPR*, 2015.

[40] X. Zhao, Q. Fei, and Q. Geng. Vision based ground target tracking for rotor uav. In *IEEE International Conference on Control and Automation (ICCA)*, 2013.

[41] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. 2014.