UAV TARGET TRACKING BASED ON PARALLEL TRACKING AND KEY FRAME DETECTION

Zhenhui WU[1], Kunpeng GE[2,*]

With the rapid application and development of unmanned aerial vehicles (UAVs), UAV target tracking has become one of the hot research directions in the field of target tracking. It is widely applied in areas such as pedestrian tracking, vehicle tracking, and obstacle avoidance. The UAV target tracker, leveraging its advantages of compact size, agile flight capabilities, and extensive coverage, effectively mitigates the limitations of traditional target tracking methods in complex environments. This paper proposes a novel multi-feature perception UAV target tracker with parallel tracking and keyframe detection to address issues such as poor robustness and low efficiency encountered by existing UAV trackers in practical application scenarios. The paper proposes a novel tracking framework comprising three main modules. The first module is a multi-feature-aware fusion tracker designed for generating predictive tracking outputs. The second module is an integral sidelobe ratio-based evaluator for parallel verification of stimulators. The third module is responsible for assessing the quality of response maps, where the ISLR (Integral Sidelobe Ratio) evaluator is employed to evaluate the response maps generated by the three single-feature trackers. The third module is a twin neural network designed for verifying detection predictions and correcting tracking results. Experiments demonstrate that, across multiple challenging unmanned aerial vehicle (UAV) image sequences, the proposed tracker featuring online two-stage evaluation with multi-cue awareness, referred to as MCVT (Multiple Cues-aware Visual Tracker with Online Two-Stage Evaluation), outperforms 20 other state-of-the-art trackers in terms of tracking accuracy, success rate, and processing time. Additionally, this multi-cue-aware tracker outperforms single-cue trackers, and the parallel tracking role played by the Siamese neural network contributes significantly to improving tracking performance.

Keywords: unmanned aerial vehicle, target tracking, parallel tracking, keyframe detection

1. Introduction

The problem of target tracking can be traced back to the appearance of the first tracking radar station SCR-28 in 1937. It wasn't until the 1970s, when the Kalman filtering theory was successfully applied in the field of target tracking, that

[1] Associate Prof., Dept. of Electronics Engineering, Yangzhou Polytechnic College, China, e-mail: zhenhuiwu@yzpc.edu.cn

[2] Eng., Dept. of Information Engineering, Suqian University, China, e-mail: KunpengGe@squ.edu.cn

the problem of target tracking gradually caught the attention of researchers and aroused widespread interest. Target tracking algorithms are widely applied in various fields such as military, agriculture, public safety, and urban development. Unmanned aerial vehicles (UAVs), referred to as “drones” in this paper, have broad applications in many areas, including pedestrian tracking [1], vehicle tracking [2], traffic monitoring [3], terrain surveying [4], obstacle avoidance [5], aerial pick-up and delivery operations [6], and aerial refueling [7]. As a type of vehicle that does not require human piloting, moves swiftly, exhibits high flexibility, and adapts well to various complex terrains, drones have become an excellent platform for implementing target tracking algorithms. In the aforementioned drone applications, visual target tracking plays a pivotal role. Fig. 1 illustrates some applications of drones, including traffic monitoring, logistics distribution, power line inspection, and terrain survey [8].

After extensive research, numerous visual trackers have been proposed in the field of unmanned aerial vehicle (UAV) tracking. However, UAV target tracking remains a challenging task primarily due to a multitude of constraining factors, including target deformation, occlusion, rotation, motion blur, rapid motion, and pose variations. Additionally, the vibration of the aircraft itself and the limited computational capabilities of the equipment also pose numerous challenges for UAV tracking, making it difficult to balance speed, robustness, and accuracy. The challenging scenarios are illustrated, where (a) depicts pose variation, (b) represents occlusion, (c) signifies deformation, and (d) indicates illumination changes:

Since the development of self-tracking algorithms, it has evolved from classical algorithms such as Meanshift [9], Particle filters [10], Kalman filtering, to later algorithms based on Correlation filtering. In recent years, artificial intelligence has garnered increasing attention, with a proliferation of deep learning-related algorithms based on various neural networks. Alongside the development of tracking algorithms, related datasets have also become rich and improved, such as OTB2015, VOT2016, UAV123, among others. These datasets provide ample basis for testing and comparing tracking algorithms.

This article aims to enhance the representational capacity of images by extracting multiple features. By employing multiple features within this tracking framework, the robustness of the tracker will be improved, enabling better adaptation to more challenging drone tracking scenarios.

2. The design of the MCVT holistic tracking framework

The overall framework of the MCVT tracker is illustrated in Fig. 1 From this figure, it can be observed that upon receiving a frame of image, the patch centered at the predicted position from the previous frame will first be extracted.

Subsequently, the MCVT tracker will extract various representative features of this patch to generate different response maps. In this project, the tracker selects to extract fHOG, CN, and grayscale features to establish three independent response maps. After obtaining these response maps, the integral sideband ratios of each map will be calculated as inputs to the integral sideband ratio evaluator. Additionally, these response maps will be normalized and fused into a fused response map. By locating the maximum value positions of fused response maps, the predicted positions can be obtained. If the integral sidelobe ratio indicates that the tracking result is reliable, then the predicted position will be output as the final result; otherwise, a region of interest centered around the predicted position will be extracted, triggering further parallel verification. The input region of interest will be validated through a Siamese neural network to determine whether it needs correction. If the input region of interest doesn't require correction or if the correction results are unreliable, then the tracker will still use the predicted position as the output. Otherwise, the tracking result will be corrected.

In Fig. 1, the green box represents the image patch used for feature extraction, with the position of the target in the (k-1)th frame as its center. The blue dashed box represents the predicted position. The blue solid line box represents the position validated by the parallel Siamese neural networks. The red dashed box represents the potential region generated in the correction stage. The red solid line box represents the corrected output position.

Each of the three individual base trackers is trained and updated using patches extracted from the previous frame image. The integral sidelobe ratio (ISLR) evaluator only requires three independent response maps. Additionally, there is a normalized preprocessing step prior to fusion. As shown in Fig. 1, the three response maps required by the ISLR evaluator also meet the requirements of the evaluator. The output of the tracker can be categorized into three scenarios: (1) Blue dotted line: The response map has passed through the ISLR evaluator, where the predicted position becomes the output; (2) Blue solid line: The response map didn't pass through the ISLR evaluator, but the image patch extracted through the predicted position passed the validation of the Siamese neural network. The output remains the predicted position; (3) Red solid line: The response map didn't pass through the integral side lobe ratio evaluator, and the image patch extracted simultaneously also didn't pass the validation of the Siamese neural network. This will activate the correction part in the parallel network, resulting in the output being the corrected result.

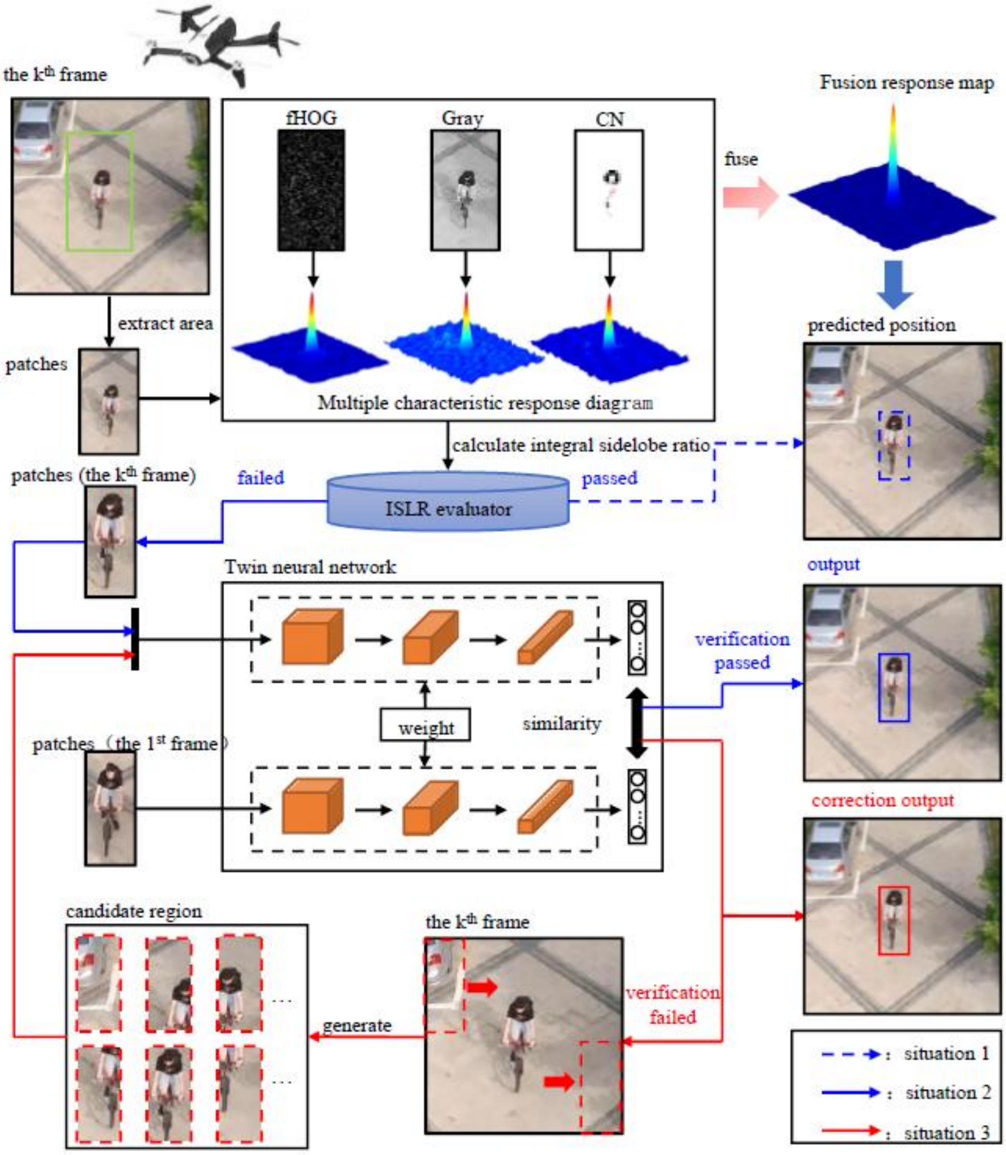

Fig. 1. MCVT tracker summary composition

The pseudocode for the MCVT tracker is depicted as shown in Table 1:

Table 1

Pseudocode for MCVT Tracker

MCVT Tracker Pseudocode

```
Input:    Target Position in Frame k-1
          3 Independent Trackers
Output:   The estimated position at frame k.
```

```
for k=2 to end do
    Extracting the search region centered around the target position from frame k-1 at frame k
    Using different features, namely fHOG, grayscale, and CN, to characterize the extracted image patches
    for Each tracker do
        Use formula ( 3.44 ) to calculate the response map
        Use formula ( 3.51 ) to calculates the response map sidelobe ratio
        Use formula ( 3.48 ) to normalize each response map
    end
    Use formula ( 3.49 ) to fuse the independent response maps.
    In the k-th frame, predicting the target position (iobj, jobj) by locating the position of the maximum value in the fused response map
    According to Fig.1, assessing the quality of the response map using the Integral Side lobe Ratio (ISLR)
    if The Integral Side lobe Ratio (ISLR) evaluator assesses the response map through ISLR evaluation then
        Using the predicted position (iobj, jobj) as the output
    else
        Call the Siamese neural network to verify the tracking results
        if Validate scores higher than the threshold $\tau 1$ then
            Use predicted position as output
        else
            Use formula (3.52) to amend the tracking results
        end
    end
    Use formula (3.43) to update individual tracking modules
end
```

3. Design of Multi-Feature Perception Tracker

The Multi-Feature Perception Tracker (MCVT) is a fused tracker designed to meet the requirements of real-time tracking, swiftly locating the target position in each frame. This fused tracker is composed of three fundamental independent trackers merged together. In this project, fDSST is utilized as the foundational tracker [11]. The distinctions among the three basic trackers depend on the different features extracted. During the process of integrating three independent response maps, the approach employed by the MCVT tracker involves utilizing the softmax formula for normalization as a preprocessing step.

3.1 Multi-feature extraction

In this multi-feature perception tracker, in order to execute the base tracker fDSST, features from different cues will be extracted, including its texture, grayscale values, and color. The features corresponding to these three cues are respectively fHOG, grayscale, and CN, denoted as $\mathcal{H}, \mathcal{G}, \mathcal{C}$ in this paper.

(1) For the HOG [12] feature, which stands for Histogram of Oriented Gradients, it is formed by computing and aggregating histograms of gradient directions over local regions of the image to form features. Because the HOG feature is extracted over local cells of the image, it exhibits invariance to local geometric and photometric transformations, as these transformations only occur at larger scales.

In this study, the tracker employs a type of HOG feature with lower dimensionality, namely fHOG. By adopting fHOG instead of HOG, the dimensionality of features decreased from 36 dimensions to 31 dimensions without significant loss of representational clues. Additionally,$\mathcal{J}^{\mathcal{H}_\mathrm{k}}$ is used to denote the tracker integrated with fHOG features and the fDSST framework in the k^th frame. Due to the fact that fHOG features consist of 31 dimensions, the first dimension is chosen here to construct the visualization of fHOG features.

(2) For grayscale features, this involves converting the RGB values of each pixel into grayscale values. The grayscale feature possesses advantages in robustness to motion blur and computational efficiency. The computation of grayscale features for each pixel is as follows:

$$\mathcal{G}_k(i,j) = \alpha_\mathrm{R} \cdot Red_k(i,j) + \alpha_\mathrm{G} \cdot Green_k(i,j) + \alpha_\mathrm{B} \cdot Blue_k(i,j) \qquad (1)$$

Here,$\mathcal{G}_\mathrm{k}(\mathrm{i},\mathrm{j})$ represents the grayscale value at position(i,j) in the k^th frame. $\mathrm{Red}_\mathrm{k}(\mathrm{i},\mathrm{j})$, $\mathrm{Green}_\mathrm{k}(\mathrm{i},\mathrm{j})$, $\mathrm{Blue}_\mathrm{k}(\mathrm{i},\mathrm{j})$respectively denote the values of the RGB channels at position(i,j).

It is worth noting that the formula employs the well-known psychophysical weights, hence in formula (1) the three weights α_R, α_G, α_B are set to 0.299, 0.587, 0.114 respectively. Moreover, $\mathcal{J}^{\mathcal{G}_\mathrm{k}}$ is used to represent the tracker integrated with the grayscale features at the k-th frame and the fDSST framework.

(3) For the CN feature, which stands for Color Names, this feature demonstrates excellent performance in image retrieval tasks. The advantage of CN features lies in their ability to effectively handle image deformations and varying shapes. Based on the RGB values of each pixel in the target, the MCVT tracker selects its color names from a predefined set of 11 basic colors. The mapping method for extracting CN features originates from [13], wherein RGB values are mapped onto an 11-dimensional color name space. The CN features are presented in the form of histograms, where the histogram reflects the number of pixels belonging to each color name.

3.2 Normalized softmax function

Softmax function, also known as the normalized exponential function, is a generalization of the logistic function in mathematics, especially in probability theory and related fields [14]. The softmax function compresses a vector of arbitrary real numbers into another vector of the same dimension, ensuring that each element falls within the range (0, 1) and the sum of all elements equals 1.

The typical expression for this function is as follows:

$$\mathrm{r_i} = \frac{\mathrm{e}^{\mathrm{V_i}}}{\sum_1^{\mathrm{C}} \mathrm{e}^{\mathrm{V_i}}} \tag{2}$$

When the softmax function is used for neural network training, here $\mathrm{V_i}$ represents the output of the preceding layer's neuron. i denotes the category index, The total number of categories is C.$\mathrm{r_i}$ represents the ratio of the index of the current element to the sum of all element indices.

To integrate the three response maps of fHOG, grayscale, and CN features, the first step is to normalize these three response maps [15]. Without this preprocessing step, the different ranges of the response maps would affect the result of the fusion. Furthermore, at the same position(i,j), if one response map exhibits a relatively high value, whereas another response map shows a significantly lower value, even approaching zero. In that case, the high value will be severely attenuated. During utilization, the softmax formula is employed for normalization as follows:

$$\mathcal{R}(\mathrm{i},\mathrm{j}) = \frac{\mathrm{e}^{\mathrm{h(i,j)}}}{\sum_{\mathrm{i}}\sum_{\mathrm{j}} \mathrm{e}^{\mathrm{h(i,j)}}} \tag{3}$$

Here, $\mathrm{h}(\mathrm{i},\mathrm{j})$ represents the pixel value of the response map at position(i,j)before normalization, $\mathcal{R}(\mathrm{i},\mathrm{j})$ is the pixel value at that position after normalization, and the sum of all pixel values after normalization equals 1.Through this approach, three separate response maps can be normalized, subsequently to be multiplied together to form a fused response map. After fusion, the resulting response map will serve as the basis for identifying the maximum response value. The fusion formula for the three individual response maps is as follows:

$$\boldsymbol{\mathcal{Q}}_{\mathbf{k}} = \boldsymbol{\mathcal{R}}^{\mathcal{H}_{\mathrm{k}}} \odot \boldsymbol{\mathcal{R}}^{\mathcal{G}_{\mathrm{k}}} \odot \boldsymbol{\mathcal{R}}^{\mathcal{C}_{\mathrm{k}}} \tag{4}$$

Here, $\mathcal{Q}_{\mathrm{k}}$ represents the fused response map.$\mathcal{R}^{\mathcal{H}_{\mathrm{k}}}$,$\mathcal{R}^{\mathcal{G}_{\mathrm{k}}}$,$\mathcal{R}^{\mathcal{C}_{\mathrm{k}}}$respectively represents three response maps computed by Equation (4). The symbol $\odot$ denotes element-wise multiplication.

Fig. 2 illustrates the difference between independent response maps and fused response maps with and without normalization operations. This demonstrates that normalization through softmax can improve the quality of response maps, thereby enhancing the accuracy of tracking results.

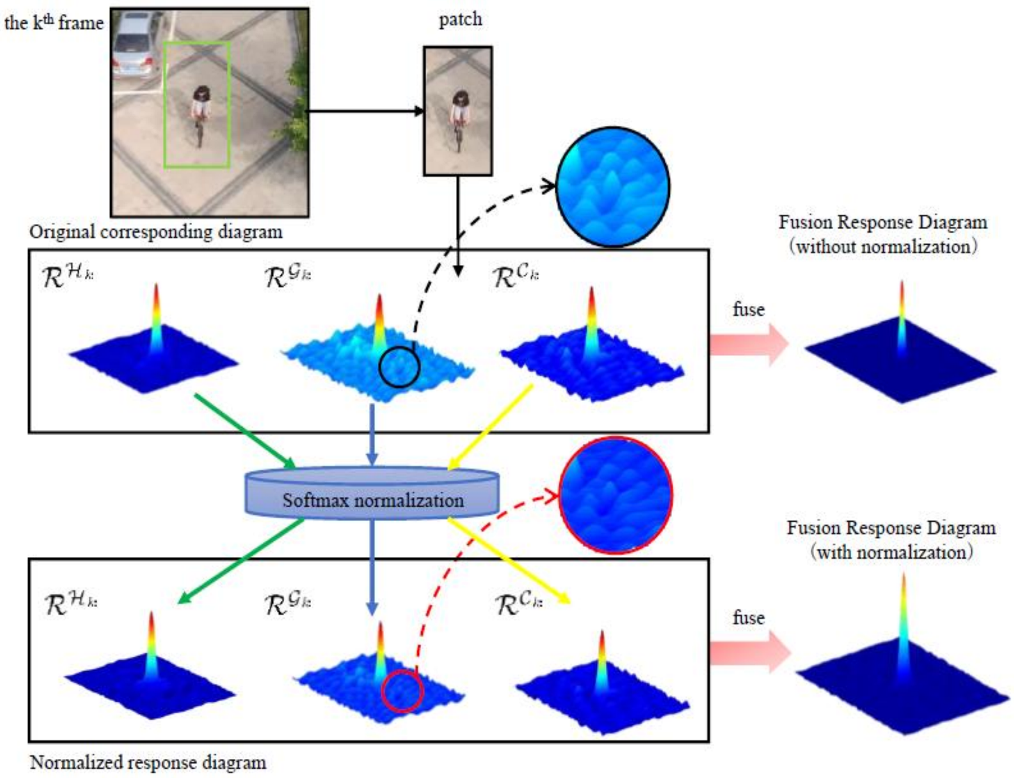
the kth frame
patch
Original corresponding diagram
$\mathcal{R}^{\mathcal{H}_k}$
$\mathcal{R}^{\mathcal{G}_k}$
$\mathcal{R}^{\mathcal{C}_k}$
Fusion Response Diagram (without normalization)
fuse
Softmax normalization
Fusion Response Diagram (with normalization)
$\mathcal{R}^{\mathcal{H}_k}$
$\mathcal{R}^{\mathcal{G}_k}$
$\mathcal{R}^{\mathcal{C}_k}$
fuse
Normalized response diagram

Fig. 2. Difference between response plots with and without normalization

From a holistic perspective, employing softmax normalization can enhance the precision of tracking [16]. Additionally, softmax normalization can suppress the negative impact brought about by negative signal values. Moreover, as mentioned earlier, normalization can eliminate differences in ranges between different response maps.

4. Design of Integral Side Lobes Ratio Evaluators

In the field of signal processing, there are numerous metrics employed to evaluate signal quality, such as Impulse Response Width (IRW), also referred to as resolution, which denotes the width of the main lobe at the 3dB drop from its peak [17]. Furthermore, Peak Sidelobe Ratio (PSLR) refers to the ratio of the maximum sidelobe to the peak height of the main lobe.

In this design, the Integrated Sidelobe Ratio (ISLR) is chosen as the evaluation metric for the response graph and serves as the triggering criterion for parallel tracking. The calculation formula for the Integrated Sidelobe Ratio is as follows:

$$\mathrm{islr} = 10\log_{10}\left\{\frac{P_{\mathrm{main}}}{P_{\mathrm{total}}-P_{\mathrm{main}}}\right\} \tag{5}$$

Here, P_{total} refers to the total energy of the response graph, and P_{main} denotes the energy of the main lobe.In the field of signal processing, the energy of the response graph is directly proportional to the square of its amplitude, expressed as $P \propto h(i,j)^2$, P denotes energy and h denotes the amplitude of the signal.

In typical scenarios, the further a signal is from its peak value, the smaller its magnitude, indicating weaker energy [18]. Therefore, in the computation process, it is unnecessary to calculate the total energy by computing the entire response graph. Instead, the value of the integral sidelobe ratio can be calculated through a simplified method. Since the response graph is a two-dimensional signal, its integral sidelobe ratio can be calculated as follows:

$$\begin{aligned}\mathrm{islr} &= 10\log_{10}\frac{\int_{-\rho_r}^{\rho_r}\int_{-\rho_r}^{\rho_r}\mathcal{R}^2(i,j)\mathrm{d}x\mathrm{d}y}{\int_{-10\rho_r}^{10\rho_r}\int_{-10\rho_r}^{10\rho_r}\mathcal{R}^2(i,j)\mathrm{d}x\mathrm{d}y-\int_{-\rho_r}^{\rho_r}\int_{-\rho_r}^{\rho_r}\mathcal{R}^2(i,j)\mathrm{d}x\mathrm{d}y}\\ &= -10\log_{10}\left[\frac{\int_{-10\rho_r}^{10\rho_r}\int_{-10\rho_r}^{10\rho_r}\mathcal{R}^2(i,j)\mathrm{d}x\mathrm{d}y}{\int_{-\rho_r}^{\rho_r}\int_{-\rho_r}^{\rho_r}\mathcal{R}^2(i,j)\mathrm{d}x\mathrm{d}y}-1\right]\end{aligned} \tag{6}$$

Here,$\mathcal{R}^2(i,j)$represents the energy of the response graph at position (i,j), ρ_r is referred to as the radius of the main lobe, which is half the width of the main lobe at the 3dB drop-off point.To simplify, the radius of a circular region with the peak position as the center and a radius of $10\rho_r$ is considered as the total energy, making the calculation of sidelobe energy more straightforward.

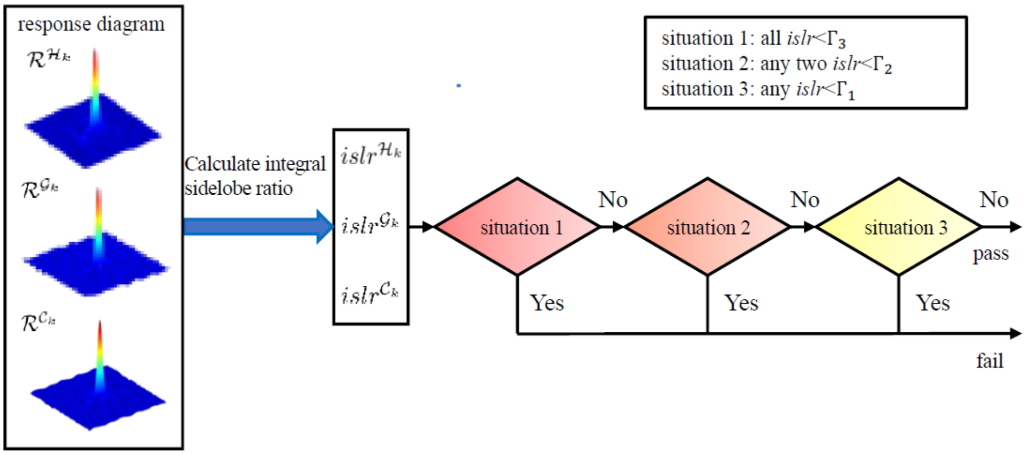
response diagram
$\mathcal{R}^{\mathcal{H}_k}$
$\mathcal{R}^{\mathcal{G}_k}$
$\mathcal{R}^{\mathcal{C}_k}$
Calculate integral sidelobe ratio
$islr^{\mathcal{H}_k}$
$islr^{\mathcal{G}_k}$
$islr^{\mathcal{C}_k}$
situation 1: all $islr<\Gamma_3$
situation 2: any two $islr<\Gamma_2$
situation 3: any $islr<\Gamma_1$
situation 1
situation 2
situation 3
No
No
No
pass
Yes
Yes
Yes
fail

Fig. 3. Framework of the Integral Paravalve Ratio Evaluator

In Fig. 3, $\mathrm{islr}^{\mathcal{H}_\mathrm{k}}, \mathrm{islr}^{\mathcal{G}_\mathrm{k}}, \mathrm{islr}^{\mathcal{C}_\mathrm{k}}$respectively represent the integral sidelobe ratio values computed from the response maps generated by trackers based on fHOG, grayscale, and CN features. The framework of this evaluator is structured into three steps for robustness considerations, as relying solely on a single integral sidelobe ratio value as a criterion, or solely on all integral sidelobe ratio values, would not be sufficiently rigorous. In the experiment, the parameter settings of the evaluator need to ensure the condition $\Gamma_1 < \Gamma_2 < \Gamma_3$. If the ratio of all three sidelobes meets the condition "all sidelobe ratios are less than Γ_3", then the evaluation result of the sidelobe ratio evaluator will be deemed as a failure; otherwise, these three sidelobe ratios will undergo further assessment. If the ratios of all three sidelobes do not satisfy the three criteria outlined in Fig. 3, then the evaluation result will be deemed as a pass, meaning the quality of the three response plots is deemed acceptable.

As shown in Fig. 4, the dashed blue box represents the output of the MCVT tracker in the absence of the integration sidelobe ratio evaluator; he dashed red box represents the output of the MCVT tracker when the integration sidelobe ratio evaluator is present, but the integration sidelobe ratio of the response plots passes evaluation. In other words, the MCVT tracker's output indicates that the integration sidelobe ratio evaluator believes that the tracking result does not require further validation or correction. The solid red box represents the output of the MCVT tracker when the integration sidelobe ratio evaluator is present, and the integration sidelobe ratio of the response plots fails the evaluation. In other words, the MCVT tracker's output indicates that the integration sidelobe ratio evaluator believes that the tracking result requires validation and correction using parallel Siamese neural networks. When the integration sidelobe ratio evaluator is adopted, inaccurate tracking results will be promptly detected, allowing them to be corrected in a timely manner by parallel Siamese neural networks.

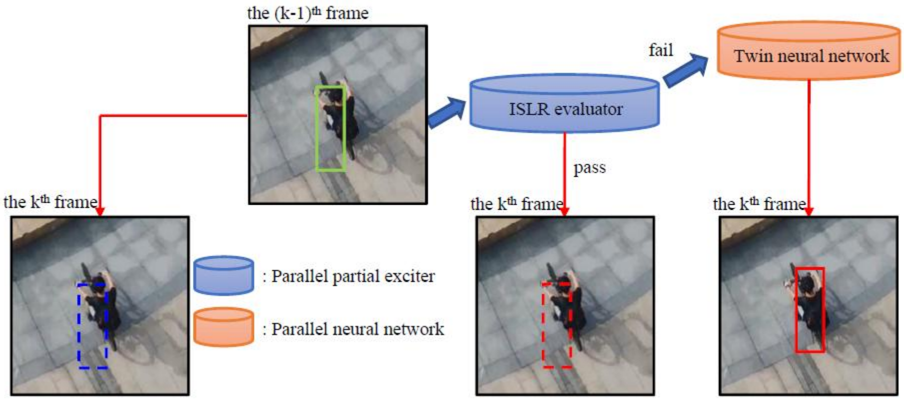

Fig. 4. Difference in tracking results with and without the integral paravalve ratio evaluator

5. Design of Siamese Neural Networks

When the response map does not pass through the side lobe ratio evaluator, the MCVT tracker will run the parallel segment of the Siamese neural network, which will be used for further validation and correction [19]. In this parallel segment, the MCVT tracker employs Siamese neural networks to validate the predicted positions generated by the fused response map [20]. If the validation result is unsatisfactory, it will be further used for refining the tracking results. The Siamese neural network is adopted here due to its characteristic of comprising two branches of convolutional neural networks, which share weights between them. Consequently, it can simultaneously process two inputs and is frequently employed for comparing image similarities. The Siamese neural network serves the purpose of measuring the similarity between two inputs by mapping them individually to a new space and representing them in that space. Subsequently, it calculates the loss function to further evaluate the similarity between the two inputs. In the semantic analysis of vocabulary, question and answer matching in Q&A, face verification, and handwriting recognition, Siamese neural networks are applied. In parallel tracking with the MCVT tracker, the first step involves inputting images from the bounding box centered around the predicted position into the Siamese neural network. These images are then compared for similarity with the images from the first frame's ground truth. If the comparison result passes (i.e., the validation score exceeds the threshold τ_1), then the predicted position will be retained as the output. Otherwise, the Siamese neural network will be invoked again for correcting the tracking result. Here, $\{R_i\}_{i=1}^{N}$ is used to denote the candidate regions generated through sliding windows, N represents the number of candidate regions. The corrected result R is determined by the following equation:

$$\mathbf{R} = \underset{\mathbf{R_i}}{\mathbf{argmax}}\, \mathbf{s}(\mathbf{T_{obj}}, \mathbf{R_i}),\ \ \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{N} \tag{7}$$

Here, $s(T_{obj}, R_i)$ returns the similarity between the tracking target T_{obj} and the candidate region R_i.

In this study, T_{obj} is extracted from the first frame of the tracked image, implying that T_{obj} remains unchanged throughout the tracking process and hence is not updated. Each correction involves finding the T_{obj} that is most similar to R_i within the bounding box of the first frame. As depicted in Fig. 5, the ISLR evaluator determines that the predicted position (illustrated by the blue dashed box in Figure 5(a) is unreliable. The parallel Siamese neural networks evaluate and correct positions within the surrounding local scope. The red dashed box in Fig. 5(b) represents the potential region generated by the sliding window. The solid red box in Fig. 5(c)represents the corrected result.

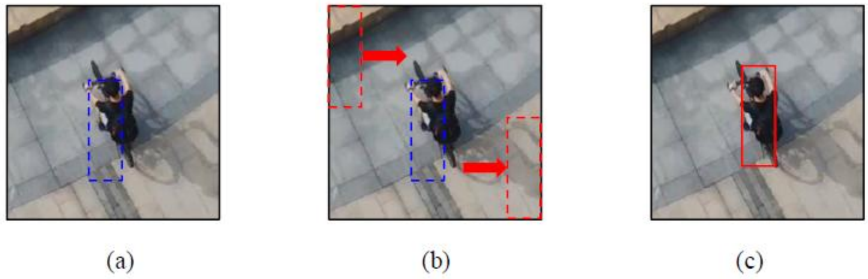

Fig. 5. Twin neural network correction process

Once the correction result is obtained, this method will further decide whether to use this result to adjust the tracking outcome. If the highest similarity exceeds the threshold τ_2, then the output is the correction result. Otherwise, the predicted position calculated by fusing response maps is retained.

6. Experimental results

The present study compares the MCVT tracker with numerous other state-of-the-art trackers, which are broadly categorized into two types for comparison: (1) Methods based on correlation filters, including CSK, KCF, fDSST, DCF, SAMF, BACF, SRDCF, MCCT, KCC, PTAV and STAPLE; (2) Other types of representation-based trackers, including IV, TLD, ASLA, Struck, MUSTER, FCT, MIL, WMIL and MEEM.

In the experiments of this study, the MCVT tracker selected for comparison is the MCCT tracker, which employs the extraction of HOG features. This is because when employing convolutional neural networks, extracting convolutional features requires a significant amount of computation. Furthermore, since the proposed MCVT tracker only utilizes handcrafted features, all other compared correlation filter-based trackers also employ only handcrafted features.

Fig. 6 illustrates success plots of precision curves for various trackers, with the testing data comprising 100 challenging UAV image sequences from UAV123.In terms of precision curves, where the threshold is set in $\xi = 20$pixels, the scores for each tracker are as follows:0.641(MCVT), 0.605(STAPLE), 0.596(MEEM), 0.591(MCCT), 0.590(BACF), 0.582(SRDCF), 0.562(PTAV), 0.551(MUSTER), 0.545(fDSST), 0.543(KCC), 0.533(Struck), 0.495(DCF), 0.483(SAMF), 0.436(TLD), 0.428(KCF), 0.416(CSK), 0.385(FCT), 0.381(ASLA), 0.347(WMIL), 0.310(IVT), 0.248(MIL).It is evident that the MCVT tracker achieved the highest score among all trackers.In terms of success rate curves, scores for each tracker are calculated based on the area under the curve. The scores for each tracker are as follows:0.449(MCVT), 0.431(MCCT), 0.426(STAPLE), 0.420(BACF), 0.418(SRDCF), 0.402(PTAV), 0.390(fDSST), 0.382(MEEM), 0.380(MUSTER), 0.380(KCC), 0.361(Struck), 0.335(SAMF), 0.318(DCF), 0.296(TLD), 0.277(KCF), 0.276(CSK), 0.260(FCT), 0.257(ASLA), 0.254(WMIL), 0.227(IVT), 0.174(MIL)Similarly to the precision curve, looking at the success rate curve, the proposed MCVT tracker still ranks first. Therefore, it is not difficult to conclude that, whether in terms of precision or success rate, the MCVT tracker outperforms the other 20 state-of-the-art trackers when compared.

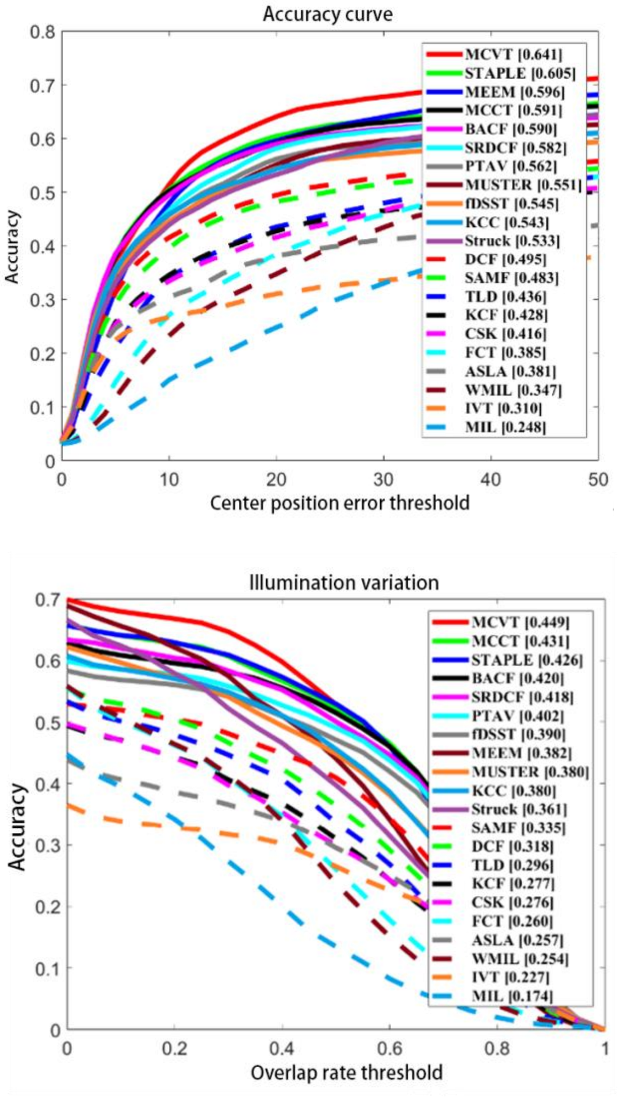

Fig. 6. Accuracy curves and success curves for all trackers on 100 challenging UAV image sequences

Besides the overall comparison in terms of precision and success rate, to better evaluate and analyze the strengths and weaknesses of tracking methods, these 100 image sequences were categorized into 12 attributes based on their tracking conditions and encountered difficulties. These 12 attributes are as follows: Illumination Variation (IV), Scale Variation (SV), Full Occlusion (FOC), Partial Occlusion (POC), Aspect Ratio Change (ARC), Similar Object (SOB), Viewpoint Change (VC), Camera Motion (CM), Fast Motion (FM), Out-of-View (OV), Background Clutter (BC), and Low Resolution (LR). By analyzing the performance of each tracker on different attributes, namely by comparing their accuracy and success rate curves across various attributes, a comprehensive analysis of the trackers can be conducted, rendering the experiments more persuasive. The ranking of scores under different attributes can also aid in determining the tracking scenarios that trackers are suited or unsuited for, thereby facilitating the analysis of their strengths and weaknesses. Table 2 and Table 3 provide detailed accuracy and success rate curves scores of each tracker across the 12 attributes. In terms of accuracy, the MCVT tracker performs the best on attributes such as IV, SV, POC, ARC, SOB, VC, CM, FM, and OV.From Table 3, it can be observed that the MCVT tracker achieved the highest scores in terms of success rate across attributes such as IV, SV, POC, ARC, SOB, VC, CM, OV, and LR. Therefore, it can be concluded that the proposed MCVT tracker in this paper demonstrates the most satisfactory performance when compared with the comprehensive evaluation of the other 20 trackers.

Table 2

Accuracy curve scores (threshold ξ = 20 pixels, top three places are marked in red, blue and green, respectively)

	IV	SV	FOC	POC	ARC	SOB	VC	CM	FM	OV	BC	LR
MCVT	52.3	58.3	44.6	56.1	54.6	70.6	57.8	61.4	48.2	54.7	48.1	50.0
MCCT	42.8	52.9	42.3	52.9	47.4	62.8	45.4	53.8	32.8	50.7	48.1	45.5
STAPLE	48.3	54.1	41.1	51.6	49.6	63.5	51.1	55.2	36.1	46.9	49.4	54.1
SRDCF	44.3	52.7	40.8	48.9	48.5	60.7	47.8	54.2	45.4	51.0	37.3	41.7
BACF	39.4	53.3	34.2	47.1	47.7	61.6	47.7	53.2	41.9	43.8	42.7	44.5
SAME	36	45.5	37.9	43.6	41.7	56.6	38.3	38.8	33.8	42.6	29.2	29.2
fDSST	42.4	49.2	37.6	47.5	45.3	60.4	43.4	45.3	35.6	48.3	33.3	43.3
DCF	32.3	43.2	31.6	38.6	36.3	53.1	39.7	39.20	23.6	34.0	32.4	38.6
KCF	28	39.0	28.7	36.5	33.2	52.6	34.5	31.9	21.7	33.7	23.6	33.5
KCC	37.2	47.9	36.0	46.3	44.0	57.4	44.2	47.1	33.8	40.8	37.2	41.5
CSK	27	38.5	29.4	33.7	31.2	47.8	31.4	32.2	25.5	33.5	22.1	33.1
PTAV	48	51.2	41.7	49.6	48.2	60.0	43.8	48.0	37.3	48.5	37 1	43.3
TLD	21.2	40.8	29.1	35.4	36.9	56.1	34.7	36.9	22.6	29.4	26.8	45.5
MUSTER	39.7	51.0	46.7	47.0	47.5	62.5	45.5	50.1	30.7	43.3	38.6	48.8
Struck	42.8	47.4	39.4	46.2	42 1	59.4	44.5	46.7	22.5	40.8	52.8	50.5
MEEM	44.2	53.2	42.9	51.5	50.7	63.1	53 1	53.6	31.1	49.6	51.0	50.2
ASLA	21.7	37.1	29.5	34.1	31.7	51.5	28.7	23 1	15.7	28.0	23.0	35.6

IVT	17.5	29.8	23.8	27.0	24.3	39.7	24.8	18 1	14.0	24.4	16.1	26.5
FCT	17.5	34.2	28.3	30.2	29.8	38.5	30.4	30.9	19.7	31.2	24.9	35.3
MIL	9.9	24.3	24.5	24.0	20.2	34.5	21.4	24.9	15.9	29.4	20.2	25.8
WMIL	16.0	30.8	26.3	28.3	27.3	38.6	28.0	30.4	19.6	29.0	27.4	35.8

Table 3

Success Rate Curve Scores (based on AUC, top three are marked in red, blue, and green, respectively)

	IV	SV	FOC	POC	ARC	SOB	VC	CM	FM	OV	BC	LR
MCVT	35.2	40.3	23.6	36.7	36.1	45.6	37.8	41.7	31.7	35.7	30.9	276
MCCT	32.1	38.3	23.6	36.1	34.1	43.6	33.5	39.1	25.0	34.3	31.7	26.5
STAPLE	34.4	37.6	22.3	34.8	34.2	41.8	36.0	39.0	25.1	32.4	33.7	25.1
SRDCF	32 1	37.5	22 1	33.4	33.7	40.4	33.7	38.8	32.1	34.2	26.1	23.3
BACF	29.2	37.2	17.6	32.2	32.7	41.4	33.1	38.6	28.9	30.4	28.5	26.0
SAME	24.8	31.1	19.8	28.4	28.5	36.6	26.9	26.6	23.3	28.0	17.4	18.6
fDSST	29.9	34.8	19.6	32.0	31.2	39.9	30.7	32.2	24.4	31.4	22.3	25.1
DCF	21.9	26.5	16.3	24.7	23.0	30.9	25.2	25.7	15.9	22.2	20.8	19.9
KCF	18.6	24.4	14.0	32.2	21.6	30.2	22.1	21.1	15.6	23.1	13.5	16.6
KCC	27.6	32.6	19. 2	28.4	29.9	38 2	31.1	32.5	21.9	26.7	25.0	227
CSK	17.5	24.8	15.4	32.0	21.0	28.7	20.2	21.3	15.0	23.9	13.5	16.3
PTAV	33.6	36.2	22.5	24.7	33.1	39.6	30.7	34.3	26.0	31.5	25.1	25.2
TLD	14.9	27.4	13.6	23.2	24.7	33.2	24.2	25.0	14.5	18.7	16.1	25.1
MUSTER	28.2	34.6	24.4	30.1	31.0	39.7	30.9	33.3	21.0	27.8	23.6	25.4
Struck	29.3	31.2	20.4	30.5	28 3	36.4	29.5	31.4	16.8	28.6	33.6	26.0
MEEM	30.2	33.2	21.9	32.7	31.5	39 1	33.4	35.0	21.9	31.5	32.2	25.3
ASLA	17.0	24.2	21.9	21.1	20.4	34.4	19.9	14.8	9.4	15.4	14.8	20.0
IVT	15.1	21.5	21.9	17.1	17.2	27.7	18.4	11.9	9.0	14.2	9.8	14.8
FCT	16.0	22.9	21.9	19.0	20.3	23.0	21.4	21.5	14.0	22.2	13.2	15.6
MIL	11.5	16.3	21.9	13.7	14.4	18.3	16.9	15.0	10.4	17.3	10.5	9.7
WMIL	11.5	21.9	21.9	18.0	19.6	22 8	21.2	21.6	15.1	20.2	16.1	15.4

Fig. 7 provides examples of three image sequences out of a total of 100, where this paper selects certain frames to visually illustrate the differences in tracking performance between the proposed MCVT tracker and the other 20 state-of-the-art trackers.

From Fig. 7, it is evident that the proposed MCVT tracker exhibits remarkable tracking performance when compared with other trackers. Even in scenarios where many other trackers fail to maintain tracking, i.e., lose the target, the MCVT tracker sustains its tracking status.

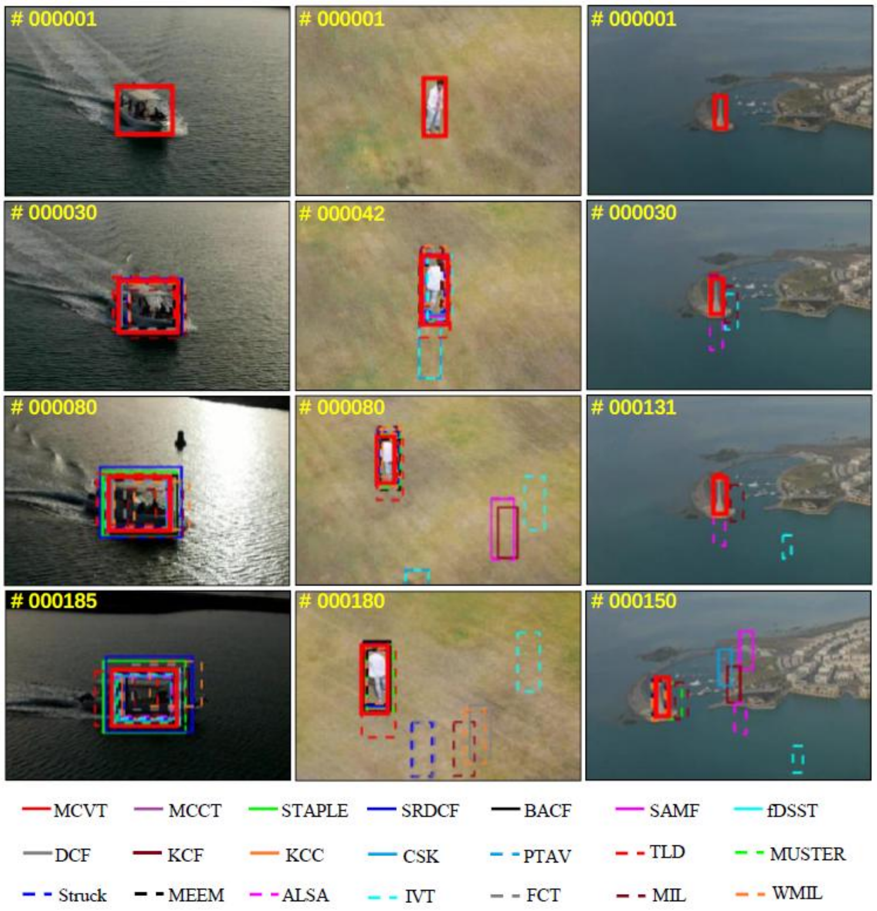

Fig. 7. Comparison of tracking results for example image sequences (from left to right image sequences are boat4, person1, building5)

Additionally, the MCVT tracker is suitable for various types of targets, such as boats, people, and buildings as shown in the above image, and it also performs well in challenging scenarios like lighting changes and fast motion.

7. Conclusion

This article proposes a novel online two-step evaluation based multi cue perceptual visual tracker, MCVT tracker, to address the problem of inefficient and low robustness tracking algorithms in many tracking applications in the field of unmanned aerial vehicles. Among them, although the ISLR evaluator increases the parameter quantity of appearance features and the computational complexity of the algorithm, it effectively utilizes the parallel computing capability of modern GPUs and does not reduce the tracking speed of the algorithm. The parallel tracking structure can improve the performance of the algorithm without requiring too much additional training, and the evaluation module based on parallel twin neural networks can maximize the performance of the algorithm. The experimental results show that the proposed method exhibits high performance on multiple standard datasets, proving its effectiveness. In future work, this algorithm can be applied to more fields that require object tracking.

REFERENCES

[1]. *N. Thakur, P. Nagrath and NHDJ. Jain*, Autonomous pedestrian detection for crowd surveillance using deep learning framework, Soft computing: A fusion of foundations, methodologies and applications, **vol.27**, no.14, 2023, pp. 9383-9399

[2]. *B. Gao, Z. Li and D. Zhang*, Roadside cross-camera vehicle tracking combining visual and spatial-temporal information for a cloud control system, Journal of Intelligent and Connected Vehicles, **vol.7**, no.2, 2024, pp. 129-137

[3]. *H. Cai, H. Lin and D. Liu*, TrafficTrack: rethinking the motion and appearance cue for multi-vehicle tracking in traffic monitoring, Multimedia Systems, **vol.30**, no.4, 2024, pp. 11-12

[4]. *J. Major*, Midair Subsurface Exploration: How UAVs solve the challenge of surveying unforgiving terrain, North American Clean Energy, **vol.15**, no.6, 2022

[5]. *X. Li, T. Jiao and J. Ma*, LSDA-APF:A Local Obstacle Avoidance Algorithm for Unmanned Surface Vehicles Base on 5G Communication Environment, Computer Modeling in Engineering and Science (English), **vol.138**, no.1, 2024, pp. 595-617

[6]. *E. Altu, ME. Mumcuoglu and I. Yüksel*, Design of an Automatic Item Pick-up System for Unmanned Aerial Vehicles, Celal Bayar Universitesi Fen Bilimleri Dergisi, **vol.16**, 2020, pp. 25-23

[7]. *D. Worth, J. Choate and J. Lynch*, Relative vectoring using dual object detection for autonomous aerial refueling, Neural Computing and Applications, **vol.36**, no.17, 2024, pp. 10143-10163

[8]. *SN. Bazha, AV. Andreev and EA. Bogdanov*, A Spatial Database of Ecosystems of the Lake Baikal Basin, Arid Ecosystems, **vol.12**, no.3, 2022, pp.243-250

[9]. *J. Li, Z. Su, ZX. Chen*, Online temperature - monitoring technology for grain storage: a three - dimensional visualization method based on an adaptive neighborhood clustering algorithm, Journal of the Science of Food and Agriculture, **vol.103**, no.13, 2023, pp. 6553-6565

[10]. *I. Maghsudlu, MR. Danaee and H. Arezumand*, Distributed state estimation with compressed and synchronized auxiliary particle filters using graph theory, Discover Electronics, **vol.1**, no.1, 2024, pp. 1-22

[11]. *K. Sun,M. He and D. Zhang*, Expression recognition algorithm based on MDS-HOG feature optimization and differential weights, Journal of Combinatorial Optimization, **vol.45**, no.1, 2023, pp. 1-17

[12]. *L. Jiang ,X. Mingwei and J. Cao*, Decentralized internet number resource management system based on blockchain technolog, Journal of Tsinghua University (Science and Technology), **vol.63**, no.9, 2023, pp. 1366-1379

[13]. *A. Gullapelly and BG. Banik*, Visual Object Tracking Based on Modified LeNet-5 and RCCF, Computer Systems Science and Engineering (English), **vol.46**, no.7, 2023, pp. 1127-1139

[14]. *K. Verma, D. Ghosh and A. Kumar,* Visual tracking in unstabilized real time videos using SURF, Journal of ambient intelligence and humanized computing, **vol.15**, no.1, 2024, pp. 809-827

[15]. *G. Riutort-Mayol, Bürkner, C. Paul and M.R. Andersen*, Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming, Statistics and Computing, **vol.33**, no.1, 2023, pp. 1-28

[16]. *J. Phillips, M. Dusseault and S. Vallado*, Investigating The Effects Of 3-Dimensional Motion Object Tracking (3D-MOT) Training On In-game College Soccer Performance, Medicine & Science in Sports & Exercise, **vol.54**, no.9S, 2022, pp. 554-554

[17]. *VV. Ponamaryov, VV. Kitov and VA Kitov*, Accounting for class hierarchy in object classification using Siamese neural networks, Computational Mathematics and Modeling, **vol.34**, no.1, 2023, pp. 27-41

[18]. *M. Chaabi, M. Hamlich and M. Garouani*, Product defect detection based on convolutional autoencoder and one-class classification, IAES International Journal of Artificial Intelligence, **vol.12**, no.2, 2023, pp. 912-920

[19]. *F. Nestle, M. Stoll and Wagner T*, Learning in high-dimensional feature spaces using ANOVA-based fast matrix-vector multiplication, Foundations of Data Science, **vol.4**, no.3, 2022, pp. 423-440

[20]. *S. Gupta, T. Mukhopadhyay and V. Kushvaha*, Microstructural image based convolutional neural networks for efficient prediction of full-field stress maps in short fiber polymer composites, Defence Technology, **vol.24**, no.6, 2023, pp. 58-82