THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 24, Number 3/2023, pp. 255–265
DOI: 10.59277/PRA-SER.A.24.3.07

TWO-LEVEL CASCADE MODEL FOR TRACKING PEDESTRIANS USING THERMAL INFRARED VIDEO INFORMATION

Xinyang BING[1], Xiaofeng MAO[2], Liying ZHENG[1], Yubo ZHANG[1], Zhongxiao LI[3]

[1] Harbin Engineering University, School of Computer Science and Technology, Harbin, China
[2] Alibaba Group, Hangzhou, China
[3] JD.com, Beijing, China
Corresponding author: Liying Zheng, E-mail: zhengliying@hrbeu.edu.cn

Abstract. Thermal infrared pedestrian tracking is a challenging task due to factors such as energy attenuation, sensor noise, occlusion, and complex backgrounds. In this paper, we design a two-level cascade model that tracks pedestrians in a thermal infrared video by the coarse-to-fine strategy to improve the tracking accuracy and success rate. The base tracker in the first level of our model is initialized and fine-tuned to get the first representation of a target which is then used to locate the target roughly. Aiming at finely locating a target, the second level consists of modality-specific part correlation filters that can capture patterns of thermal infrared pedestrians. The outputs of part correlation filters are aggregated together by normalized joint confidence that can effectively suppress low confidence predictions to make a final decision. We adaptively update each part filter by a weighted learning rate and accurately estimate pedestrian scale by a scale filter to improve tracking performance. The experimental results on the PTB-TIR benchmark show that the proposed cascade tracker further emphasizes crucial thermal infrared features. Thus it can effectively relieve the problem of object occlusion. Our experimental results show the superiority of the proposed tracker over the state-of-the-art trackers, including SRDCF, GFS-DCF, MCFTS, HDT, HCF, MLSSNet, HSSNet, SiamFC_tir, SVM, and L1APG.

Keywords: location estimation, normalized joint confidence, object tracking, part correlation filter, thermal infrared pedestrian.

1. INTRODUCTION

Thermal InfraRed (TIR) pedestrian tracking is an important computer vision task with broad applications to surveillance and early warning. TIR videos are generally characterized by the following two features. First, the pixel intensities in a TIR video depend on the radiant temperatures of objects rather than reflected lights as in common visual videos. Second, a TIR video has much fewer details than a visible one due to its low resolution. Specifically, a TIR video generally lacks texture, color, and even structural information of an object. Therefore, traditional hand-crafted features such as Histogram of Oriented Gradients (HOG) [1] and Color Names (CN) [2] may lose their ability to effectively represent the appearance of an object, and thus their discrimination of different objects in the TIR domain is low [3]. TIR and visible tracking is compared in [4], concluding that well-designed features for TIR spectrum play a crucial role.

In the past few years, numerous TIR pedestrian tracking models have been reported. Early TIR pedestrian trackers are based on motion tracking techniques such as Kalman filtering [5] and particle filtering [6, 7]. For example, Xu et al. [5] combined a Kalman filter with mean-shift clustering to detect and track pedestrians in videos obtained by a single night-vision camera installed on a vehicle. Wang et al. [6] tracked TIR pedestrians in a particle filter framework that fuses multi-features. After pre-processing features in thermal images with Wigner distribution, a particle filter-based motion tracker is proposed by Padole and Alexandre [7]. The abovementioned simplicity trackers lead them to suffer from background clutters. Thus, researchers also study some complex techniques to improve the performance of TIR trackers further. For example, [8] estimates TIR target position with the fluctuation of the relative change rate between two adjacent frames. [9] tracks TIR targets by using non-local dynamic pattern matching.

Over the last few years, deep architectures [10-12] have brought impressive advances in computer vision tasks. Deep models are now trending ones in object tracking [13-17], and so does in TIR pedestrian tracking. Deep trackers can capture the general characteristics of an object in different video frames and are robust to external interferences such as illumination, occlusion, and motion blur. One type of the most popular deep trackers is Siamese-network-based trackers [18-22]. Recently, Liu et al. [19] proposed a video prediction network to update the TIR pedestrian tracking template of SiamRPN [18]. Zheng et al. [20] proposed a real-time TIR pedestrian tracker which combines a CNN-based prediction model with SiamRPN [18] to improve the tracking performance. Though these Siamese-network-based trackers [18-22] achieve better tracking performance than traditional ones, they are trained end-to-end and thus consume a large of computing resources. To solve such problems, Correlation Filters (CFs) are introduced to deep trackers. Ma et al. [14] first proposed to use hierarchical convolution to learn and express the tracking target. They improve the tracking accuracy by learning CFs in different layers. Liu et al. [23] proposed a CF-based ensemble tracker with multi-layer convolutional features for TIR tracking. But the tracking ability of such CF-based methods doesn't satisfy tracking objects' needs in severe occlusion scenes.

The main idea of cascade CNN[24] is to gradually optimize from coarse to fine. The deformable part model [25] includes a global root filter and several component models. Each component model consists of a spatial model and a component filter. Sevilla Lara et al. [26] first proposed the object representation method Distribution Fields (DFs) for object tracking. And in [27] compared trackers based on spatial structure features with ones based on DFs and concludes that DFs are more suitable for TIR tracking. Motivated by the cascade CNN and the deformable part model, this paper proposes a two-level cascade model in a CF framework [28–32] for TIR pedestrian tracking. At the first level, the base tracker uses both deep features and hand-crafted ones to get the initial representation of a target which is then used to locate the target roughly. To refine these initial estimations, our model sets up the second level that consists of modality-specific filters capturing low-level TIR patterns.

The contributions of this paper are summarized as follows:

1) Compared with existing TIR tracking methods, we designed a two-level cascade framework for tracking TIR pedestrians from coarse to fine. The base tracker in the first level is used for rough localization of the target, while the second level employs a modality-specific fine estimation module that captures patterns of thermal infrared pedestrians to improve accuracy.

2) We proposed a novel fine estimation module. Firstly, we construct part filters based on the DFs using the rough position information provided by the base tracker. Then, we design a normalized joint confidence to aggregate the output of each part filter. The final position and scale of the target are obtained through a maximum aggregation response and scale estimation module.

3) We developed an update strategy that preserves historical information for each part filter, effectively mitigating drift problems. Specifically, it improves the accuracy and success rate of tracking in occlusion scenarios.

4) We conducted extensive experiments on the PTB-TIR [33] dataset, a benchmark for TIR pedestrian tracking, using 60 challenging sequences to demonstrate the effectiveness of the proposed model and achieved promising results.

The rest of this paper is organized as follows. Section 2 provides a detailed description of the proposed two-level cascade tracker. Section 3 explains the evaluation procedure of the proposed tracker, and presents the experimental results and some analysis. Section 4 gives some conclusions.

2. THE PROPOSED TWO-LEVEL CASCADE TRACKER

Figure 1 shows the proposed two-level cascade tracker. The first level of our model initializes and fine-tunes a base tracker followed by inputting a TIR video frame. Then the base tracker roughly locates the target pedestrian. Next, the fine location of the target is estimated by the second level termed as fine estimation module. Here, the base tracker usually is a tracker performing well to track the video sequence frame by frame. As shown in Fig. 1, the fine estimation module first divides the target into n components. Then it uses DFs based part filters to calculate the normalized joint confidence which is used by the scale estimation module to estimate the target scale. Finally, the final tracking decision is made with the bagging process.

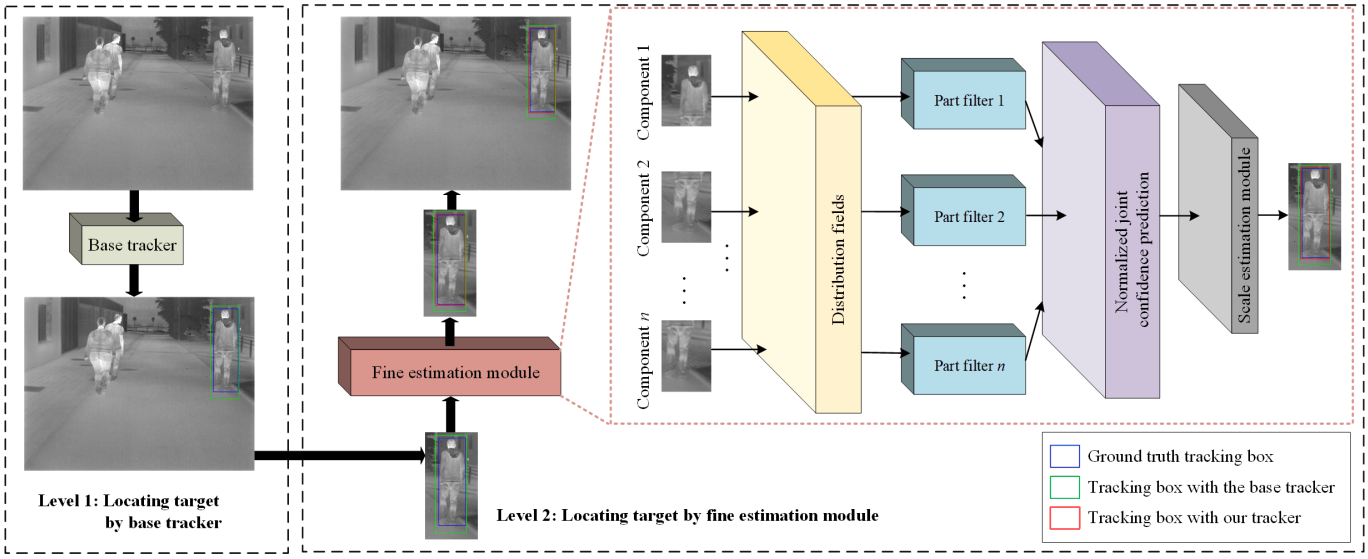

Fig. 1 – The framework of the proposed tracker.

2.1. Distribution fields

Here we use DFs to model the appearance of a TIR pedestrian target. A target is first decomposed into n components from which local DF features are extracted. The smoothed DFs are constructed by (1).

$$\mathrm{dfs}(I) = \mathrm{dfs}\big(I(i,j),k\big) = \delta\big(I(i,j),k\big) * h_{\sigma_s} * h_{\sigma_f} \tag{1}$$

where $\mathrm{dfs}\big(I(i,j),k\big)$ is the smoothed DFs of video frame I at the i^{th} row and the j^{th} column, and k indicates the possible value of pixel $I(i,j)$. $\delta(\cdot)$ is Kronecker delta function. h_{σ_s} is a 2D Gaussian kernel with the standard deviation σ_s, h_{σ_f} is a 1D Gaussian kernel with the standard deviation σ_f along the feature dimension. '$*$' is the convolution operator.

Suppose $c_1, c_2, \cdots, c_n$ to be the n components of the target. The DFs of each component can be viewed as a local radiant temperature description of the target. Thus, our model adopts $\left\{\mathrm{dfs}(c_v) \middle| v \in \{1,2,\cdots,n\}\right\}$ which are obtained by (1) as TIR-specific features of a target.

2.2. Part correlation filters

CFs for visual tracking has attracted considerable attention due to their high computational efficiency using the fast Fourier transform. In this work, we use Kernel Correlation Filter (KCF) [30] to build part CFs.

Let $\mathbf{X}_t^v = \mathrm{dfs}(c_v) \in \mathbb{R}^{O\times Q\times D}$ denote the DFs extracted from the v^{th} component of an object in the t^{th} frame. After circularly shifting $\mathbf{X}_t^v$ along O and Q dimensions, we get shifted samples $\mathbf{X}_t^v(o,q)$, where $o \in O$ and $q \in Q$. By assigning a Gaussian label $g(o,q) = e^{-\frac{(o-O/2)^2+(q-Q/2)^2}{2\sigma^2}}$ to each $\mathbf{X}_t^v(o,q)$ (here σ is the kernel width), we get training samples $\left\{\mathbf{X}_t^v(o,q), g(o,q)\right\}$. The goal of training is to determine the filter weights by optimize objective function (2):

$$\min_{\mathbf{w}_t^v} \sum_{o,q} \left(f\left(\mathbf{X}_t^v(o,q)\right) - g(o,q)\right)^2 + \lambda \left\|\mathbf{w}_t^v\right\|^2 \tag{2}$$

where λ is a regularization parameter, and $f(\mathbf{X}_t^v(o,q)) = (\mathbf{w}_t^v)^{\mathrm{T}} \mathbf{X}_t^v(o,q)$ is the response of the filter. By using the kernel trick and Fourier convolution theorem, the optimal $\mathbf{w}_t^v$ of (2) can be expressed as $\mathbf{w}_t^v = \sum_{o,q} \alpha_t^v(o,q)\kappa(\mathbf{X}_t^v(o,q))$ with (3):

$$\alpha_t^v = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(g)}{\mathcal{F}(\kappa(\mathbf{X}_t^v(o,q)), \mathbf{X}_t^v) + \lambda}\right) \tag{3}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote Fourier transform and its inverse transform, respectively. κ is a kernel function. After training, we have the appearance representation in Fourier domain $\mathcal{F}(\mathbf{X}_t^v)$ and the filter coefficient in Fourier domain $\mathcal{F}(\alpha_t^v)$. They are together formed a KCF [30] tracker which is served as a part correlation filter in our tracker. The response of each part filter to sample $\mathbf{X}_t^v$ is computed by (4):

$$P_t^v = \mathcal{F}^{-1}(\mathcal{F}(\kappa(\mathbf{X}_t^v, \mathbf{X}_{t-1}^v)) \odot \mathcal{F}(\alpha_t^v)) \tag{4}$$

where $\odot$ is the element-wise product.

2.3. The final location of a target

To infer the final position of a target, we modify the aggregation strategy in [31] by considering both the value and the position of the maximum response of each part filter. For each component of a target pedestrian, we run its corresponding part correlation filter and get its part response map. The Peak-to-Sidelobe Ratio (PSR) [31] confidence score of each part response map is computed by (5):

$$\mathrm{PSR}_t^v = \frac{\max(P_t^v) - \mu_{P_t^v}}{\sigma_{P_t^v}} \tag{5}$$

where $\mu_{P_t^v}$ and $\sigma_{P_t^v}$ are the mean value and the standard deviation of the v^{th} part response map, respectively. PSR_t^v can be used to quantify the sharpness of the correlation peak, and the high value of PSR_t^v means the high matching score of the current frame to the previous frames. Besides the PSR confidence score, we also consider the Maximum Offset Constraint (MOC). The maximum response position of a part tracker drifts as time goes on and error is accumulated. To alleviate such drifting, we construct MOC in (6):

$$\mathrm{MOC}_t^v = \exp\left(\frac{-\left\|\wedge_t^v - \Delta_t^v\right\|_2^2}{2\sigma_v^{\,2}}\right) \tag{6}$$

where $\wedge_t^v$ and Δ_t^v denote the prediction shift of a base tracker and that of a part tracker for the v^{th} component at t^{th} frame, respectively. σ_v controls the regular size of the maximum offset. MOC_t^v keep the final prediction near the original result, and thus dominates the drifting level.

As given by (7), PSR_t^v together with MOC_t^v is used to compute the normalized joint confidence score:

$$\mathrm{Conf}_t^v = \frac{\mathrm{PSR}_t^v \times \mathrm{MOC}_t^v}{\sqrt{\sum_{v=1}^{n} \mathrm{PSR}_t^v \times \mathrm{MOC}_t^v}} \tag{7}$$

The final location of a target is obtained by computing the maximum response position (x^{final}, y^{final}) of two-level cascade model, as given by (8):

$$(i_t^{final}, j_t^{final}) = \arg\max_{i,j}(\mathrm{Conf}_t^v P_t^v) \tag{8}$$

2.4. Model updating

Our method adopts a CF as a part tracker. Since a correlation filter is sensitive to imaging conditions and appearance changes, the feature template of a target component should be continuously updated during tracking. Similar to traditional methods, we online update the classifier coefficients with simple linear interpolation. To decrease the influence of the responses of part trackers with low confidence, we adaptively update each part filter with an independent weighted learning rate. Similar to [31], the weights are controlled by a confidence score and a threshold. The appearance $\mathcal{F}(\mathbf{X}_t^v)$ of a target is updated by (9). Accordingly, the correlation filter coefficient $\mathcal{F}(\alpha_t^v)$ is also updated by a similar method.

$$\mathcal{F}(\mathbf{X}_t^v) = \begin{cases} (1-\eta \mathrm{Conf}_t^v)\mathcal{F}(\mathbf{X}_{t-1}^v) + \eta \mathrm{Conf}_t^v \mathcal{F}(\mathbf{X}_t^v) & \text{if } \mathrm{Conf}_t^v > \text{threshold} \\ \mathcal{F}(\mathbf{X}_{t-1}^v) & \text{otherwise} \end{cases} \tag{9}$$

where η is a designated learning rate which is further weighted by confidence score Conf_t^v in our method.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1. Implementation details and evaluation criteria

We implement the proposed cascade model using MATLAB 2016b on an Intel(R) I7-10700 CPU and an NVIDIA GeForce RTX 3080 GPU. We use PTB-TIR [33] dataset as the TIR pedestrian tracking benchmark. The dataset includes 60 TIR sequences with annotations. Each sequence has ten attribute labels.

Three trackers, i.e., GFS-DCF [34], HCF [14], and MCFTS [23] are respectively selected as the base tracker in our model. For the chosen base tracker, GFS-DCF [34] uses both depth features and hand-crafted features. HCF and MCFTS only use depth features. Particularly, MCFTS [23] is dedicated to TIR tracking. The intensity channel is not used for initialization and fine-tuning of GFS-DCF to speed up the tracking. Since GFS-DCF is an RGB tracker, we adapt the following parameters to present the different characters of TIR data. Following [23], the cell size of HOG features is 4. The configurations of HCF [14] and MCFTS [23] are the same as those reported in [14] and [23]. To construct DFs, we follow the parameters recommended in the [27]: Intensity bins of DFs is fixed as 16, spatial smoothing parameters σ_s and intensity smoothing parameter σ_f in (1) is [2,1] and 0.625 respectively. For each part CF, we use the same parameters as KCF [30]: regularization parameter in (2) and (3) λ is 10^{-4}, threshold and Learning rate η in (9) is 0.1 and 0.01 respectively. The regular size of the maximum offset σ_v in (6) is set to 5. How to optimally divide an object into *n* components is still an open problem. For simplicity, we directly divide the bounding box of a target pedestrian into *n* equal parts, with *n* being the square of a positive integer. We choose the scale filter of DSST [35] as the scale estimation module, and GFS-DCF [34] uses its own scale model.

We use One-Pass Evaluation (OPE) plots to show the precision and success rate of trackers. A precision plot measures the precision over a range of thresholds and we report the precision score within a given threshold (20 pixels) in the legend. A success plot shows the ratios of successful frames at the threshold varying from 0 to 1 and the Area Under the Curve (AUC) of success rate will be given in its legend. More details of evaluation criteria can be found in [33]. Algorithm 1 gives a brief outline of our two-level cascade model. In all experiments, we report the average result over 3 runs.

Algorithm 1 summarizes the main steps of the proposed tracking model.

Algorithm 1 Proposed tracking model: iteration at frame *t*.

Inputs: Video frame I_t, previous target position (i_{t-1}, j_{t-1}), scale s_{t-1}.

Outputs: Estimated object position (i_t, j_t), scale s_t.

Level 1: roughly locate the target position:

Estimated object position (i_t, j_t) and scale s_t using base tracker.

Level 2: fine estimation module:

1: Extract target area in frame t using **Level 1** estimated target position (i_t, j_t) and scale s_t.

2: Divide the target area into n components and extract the corresponding sub-target area for each component c_v.

3: **Position estimation:**

foreach component c_v **do**:

Extract smoothed DFs $\mathbf{X}_t^v$ using (1) and computing the corresponding response P_t^v using $\mathbf{w}_{t-1}^v$ and (4).

Compute the normalized joint confidence score Conf_t^v based on (7).

Coarse-to-fine estimate the final object position $(i_t, j_t) = (i_t^{final}, j_t^{final})$ using (8).

4: **Scale estimation**:

Estimated object scale s_t using scale model DSST [35].

5: **Model update:**

foreach part correlation filters **do**:

Update appearance model $\mathcal{F}(\mathbf{X}_t^v)$ using (9).

Update scale model.

3.2. The performance of fine estimation module

In this section, we cascade the proposed fine estimation module with the abovementioned three base trackers to evaluate the influences of the fine estimation module on the tracking.

3.2.1. Tracking speed

First of all, we analyse the influence of *n* on tracking speed. On our platform, the OPE speed of GFS-DCF [34], HCF [14], and MCFTS [23] are respectively 7.7 FPS, 29 FPS, and 7 FPS, while the tracking speed of cascade trackers with a different number of target components are listed in Table 1. Compared to the speed of the corresponding base tracker, we can see that our fine estimation module slows down the tracking speed. The tracking speed is gradually decreased with increasing *n*. The deceleration is extremely significant when *n* is greater than 9. Thus, in practice, *n* can be 4 or 9. To balance speed and performance, we set *n* to 4 in following experiments.

Table 1

The tracking speed of cascade trackers

Trackers	1-Speed(FPS)	4-Speed(FPS)	9-Speed(FPS)	16-Speed(FPS)
Cascade_GFS-DCF	5.8	5.2	4.3	3.6
Cascade_HCF	12.0	9.5	7.0	5.3
Cascade_MCFTS	5.3	4.8	4.1	3.4

3.2.2. Tracking precision and success rate

Figure 2 shows the precision and success plots of cascade trackers and base trackers. The cascade trackers are highlighted in bold. The precision plots suggest that proposed cascade methods provide an absolute gain of 1.7%, 10%, and 11.5% improve accuracy over base methods GFS-DCF [34], MCFTS [23], and HCF [14], respectively. The success plots show the success rate overall of 60 sequences. From the resulting plots, we can see that our cascade trackers achieve AUC of 0.598, 0.527, and 0.550 for the base tracker GFS-DCF [34], MCFTS [23], and HCF [14], respectively. Such AUCs significantly outperform corresponding base trackers.

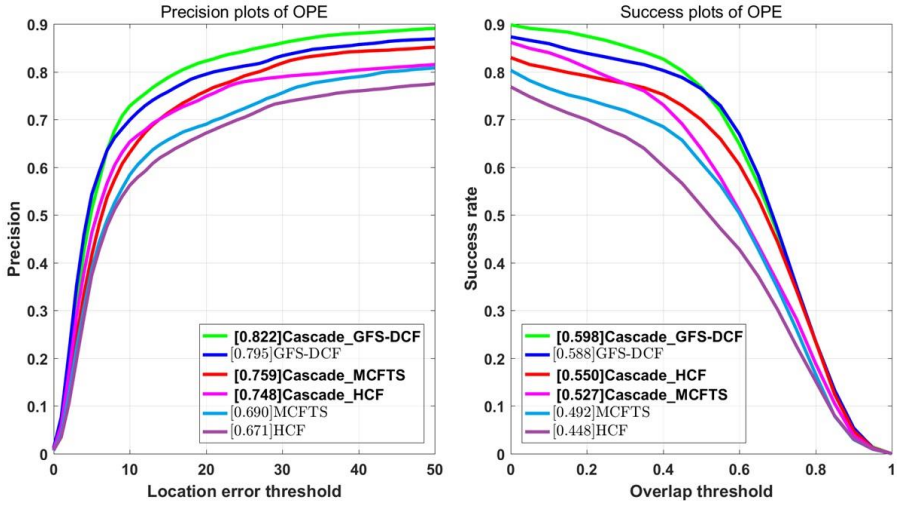

Fig. 2 – The overall performance of the trackers using OPE plots.

In addition, we find that the boosting effects of our fine estimation module on tracking depend on the type of the base tracker. The performance improvement of Cascade_HCF is more considerable than Cascade_MCFTS and Cascade_GFS-DCF. Such different improvements may be caused by the different target features used in the base trackers. MCFTS [23] uses more depth features than HCF [14]. GFS-DCF [34] uses both hand-crafted and depth features, having a good discrimination ability to avoid drifts. Compared to base trackers, the corresponding cascade tracker improves the tracking performance to a certain degree, proving that our tracker uses more suitable features for characterizing a TIR pedestrian than the base tracker. Adding batch normalized weights and adaptive updating strategy, the drift problem that is one of the most challenging problems besetting tracking tasks is further relieved. As a result, the overall tracking performance is improved.

3.2.3. Attributes analysis

To further evaluate the strength and weakness of the proposed model, we evaluate ten common attributes in PTB-TIR dataset: DeFormation (DF), OCClusion (OCC), Scale Variation (SV), Background Clutter (BC), Low Resolution (LR), Fast Motion (FM), Motion Blur (MB), Out-of-View (OV), Intensity Variation (IV), and Thermal Crossover (TC). The AUC of success rate and the precision score on attribute subsets are listed in Table 2 and Table 3, respectively. The best results are highlighted in bold. We can see that the proposed cascade tracking strategy mainly improves the performance on DF, OCC, and BC subsets. For Cascade_GFS-DCF, the tracking accuracy and tracking success rate are improved for the attributes DF, OCC, and BC. Cascade_MCFTS performs better on the attributes of DF, OCC, BC, LR, FM, OV, MB, and SV. Cascade_HCF has a better tracking success rate than other trackers on all attribute subsets, and its tracking accuracy is better than others except for attribute IV. Figure 3 shows several tracking results of Cascade_GFS-DCF and GFS-DCF [34] for visually demonstrating the performance of our tracker. The overlap rates are plotted to assist qualitative analysis. The corresponding cascade tracker performs best for each selected base one on the three attributes: DF, OCC, and BC. Therefore, detailed analysis on these three attributes is given.

Table 2

The AUC of success rate on ten attribute subsets

Trackers	Attribute									
	DF	OCC	BC	IV	LR	FM	MB	OV	SV	TC
Cascade_GFS-DCF	**0.598**	**0.577**	**0.589**	0.445	0.575	0.669	0.583	0.635	0.583	0.567
GFS-DCF	0.588	0.556	0.570	**0.457**	**0.655**	**0.688**	**0.623**	**0.653**	**0.594**	**0.598**
Cascade_MCFTS	**0.527**	**0.514**	**0.513**	0.342	**0.530**	**0.633**	**0.501**	**0.545**	**0.501**	0.538
MCFTS	0.492	0.501	0.483	**0.385**	0.418	**0.553**	0.444	0.537	0.465	**0.558**
Cascade_HCF	**0.550**	**0.533**	**0.537**	**0.364**	**0.568**	**0.692**	**0.536**	**0.532**	**0.532**	**0.560**
HCF	0.448	0.449	0.448	0.327	0.375	0.486	0.417	0.391	0.391	0.470

Table 3

The precision score on ten attribute subsets

Trackers	Attribute									
	DF	OCC	BC	IV	LR	FM	MB	OV	SV	TC
Cascade_GFS-DCF	**0.822**	**0.782**	**0.799**	0.460	0.884	0.974	0.841	0.764	0.781	0.792
GFS-DCF	0.795	0.747	0.765	**0.546**	**0.935**	**0.986**	**0.851**	**0.797**	**0.795**	**0.819**
Cascade_MCFTS	**0.759**	**0.726**	**0.742**	0.277	**0.882**	0.880	**0.746**	**0.699**	**0.718**	**0.796**
MCFTS	0.690	0.683	0.669	**0.310**	0.732	**0.806**	0.658	0.614	0.660	0.776
Cascade_HCF	**0.748**	**0.715**	**0.730**	0.395	**0.894**	**0.935**	**0.786**	**0.623**	**0.718**	**0.795**
HCF	0.671	0.657	0.662	**0.443**	0.696	0.762	0.656	0.519	0.618	0.734

Attribute DF. This subset consists of three challenging TIR sequences that have DF properties with pedestrians deform in motion. For example, in the 458th frame of the sequence "crossing", the tracking pedestrian rotates body during moving. In the 426th frame of the sequence "crouching" and the 110th frame of the sequence "street2", the limbs and trunk of the pedestrian deform during walking. Since GFS-DCF [34] cannot accurately characterize a TIR target under such deformation circumstances, its tracking performance degrades on this subset.

Attribute BC. The video backgrounds in this subset contain similar pedestrians to the target and(or) complex context. For example, background interferences appear around the tracking target in the 458th frame of the sequence "crossing". In the 426th frame of the sequence "crouching" and in the 110th frame of the sequence "street2", several fake pedestrians appear around the true one. GFS-DCF [34] suffers from serious tracking drifts, resulting in wrong prediction in the case of such BC occurring. Our cascade tracker adopts batch normalized confidence and adaptive online template updating strategy. As a result, it can more effectively suppress tracking drifts and is more robust to BC interferences than the base tracker.

Attribute OCC. OCC is caused by the tracking target being obscured by unexpected objects. For example, in the 426th frame of the sequence "crouching" and the 110th frame of the sequence "street2", the target is occluded in the normal motion process. For sequence "crouching", most parts of the target are occluded by other objects. On this subset, GFS-DCF [34] drifts slowly, and eventually, it tracks a wrong target. On sequence "street2", GFS-DCF [34] drifts rapidly since the target is completely occluded. Above results indicate that GFS-DCF [34] is insufficient to deal with the sudden changes of occlusion. Our model adds TIR features to the features extracted by a base tracker, resulting in strong discrimination ability and strong robustness to the appearance changes. Together with the strategy of batch normalized weight and adaptive online model updating, our tracker can accurately predict the overall position. Thus it can effectively track a target under occlusion. Moreover, from the overlap rates, we can see that the tracking box of Cascade_GFS-DCF is closer to the ground truth, which is beneficial to long-term tracking.

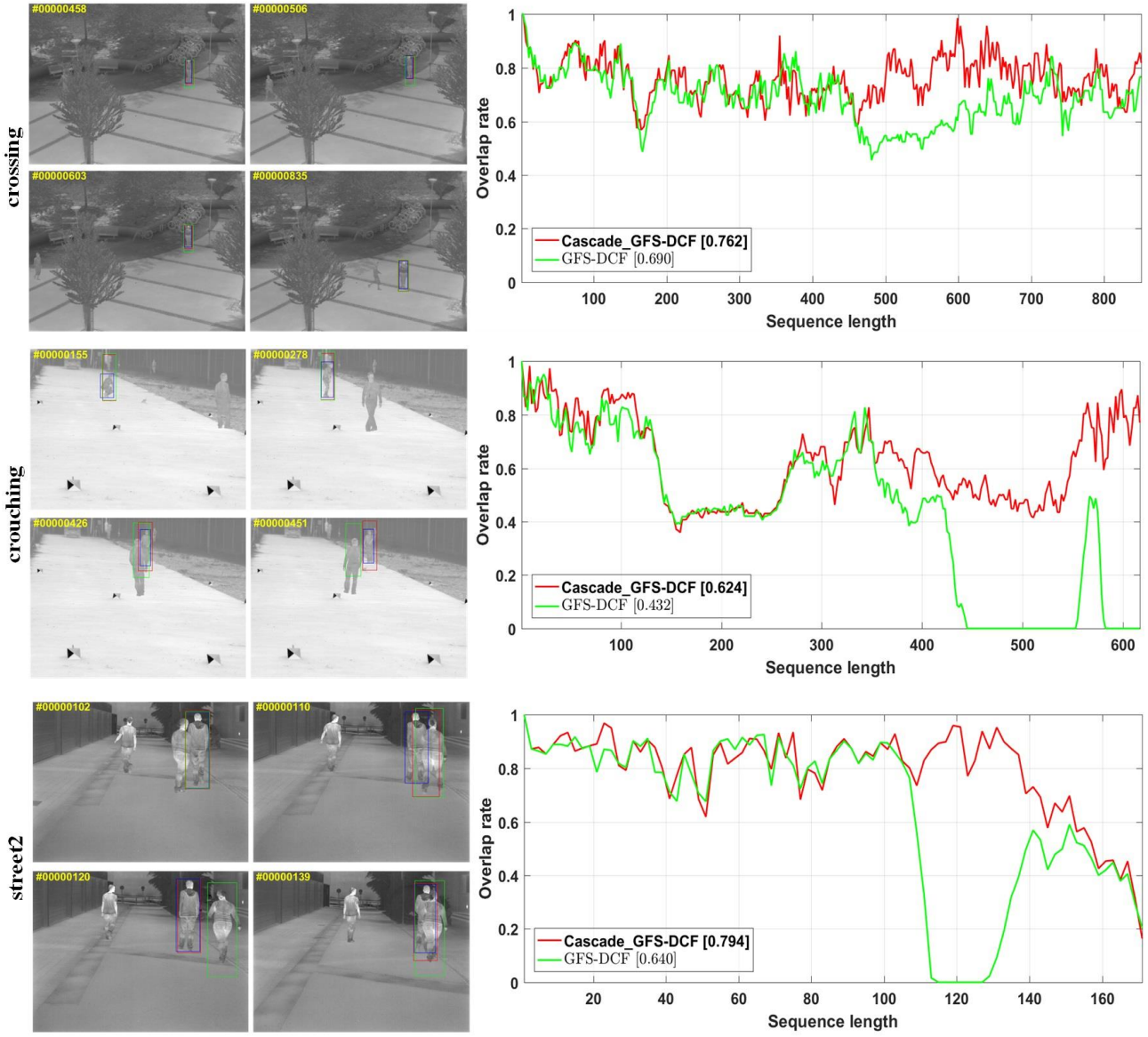

Fig. 3 – Tracking results on three challenging TIR sequences.

3.3. Comparison to other trackers

We compare our two-level cascade tracker to 10 state-of-the-art trackers that cover a wide range of implementation approaches on PTB-TIR. The 10 compared trackers consist of five CF-based trackers: SRDCF [36], GFS-DCF [34], MCFTS [23], HDT [16], and HCF [14]. Three Siamese-network-based trackers, including MLSSNet [37], HSSNet [38], and SiamFC_tir [39]. Two other trackers: SVM [40] and sparse representation based tracker L1APG [41]. For more details about these trackers, please refer to the corresponding literature. Figure 4 visualizes comparison results. In Fig. 4, the farther away from the circle's center, the better performance a tracker achieves. We can see that Cascade_GFS-DCF obtains the best precision score and the AUC of success rate among all compared trackers, demonstrating the good performance of the two-level cascade tracker for predicting the position of TIR pedestrian.

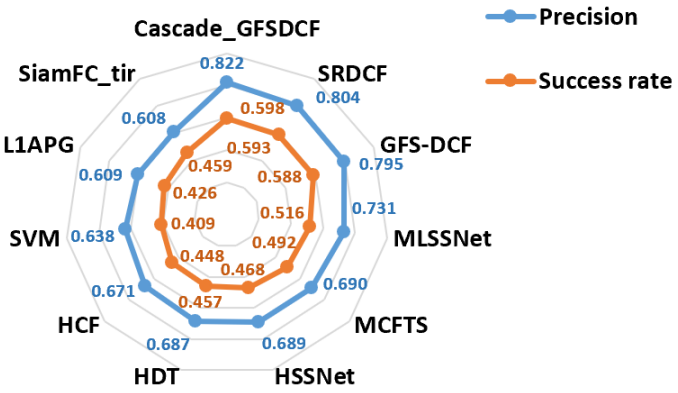

Fig. 4 – Comparison of the proposed tracker with other ones on PTB-TIR datasets.

4. CONCLUSIONS

In this work, a two-level cascade model for tracking TIR pedestrians has been designed. Unlike the previous trackers, we introduce a general cascade framework to TIR target tracking, dividing the tracking model into two levels: coarse location estimation based on base tracker and adaptive fine estimation module based on TIR details. Using the adaptive weighting update strategy and normalized joint confidence, our tracker is robust to occlusion, background clutter, and appearance changes.

The experimental results show that our method has a better tracking performance on the TIR sequence than the base trackers in tracking precision and success rate. However, TIR tracking remains a challenging task due to the complex scenes. In the future, we are going to focus on improving the tracking speed and exploring explainable TIR features.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61771155.

REFERENCES

1. N. DALAL, B. TRIGGS, *Histograms of oriented gradients for human detection*, 2005 IEEE Computer Society Conference on Computer Vision and Pattern recognition, 2005, pp. 886–893.
2. M. DANELLJAN, F. SHAHBAZ KHAN, M. FELSBERG, J. VAN DE WEIJER, *Adaptive color attributes for real-time visual tracking*, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1090–1097.
3. E. GUNDOGDU, A. KOC, B. SOLMAZ, R.I. HAMMOUD, A. AYDIN ALATAN, *Evaluation of feature channels for correlation-filter-based visual object tracking in infrared spectrum*, 2016 IEEE Conference on Computer Vision and Pattern recognition Workshops, 2016, pp. 24–32.
4. E. GUNDOGDU, H. OZKAN, H. SECKIN DEMIR, H. ERGEZER, S. KUBILAY PAKIN, *Comparison of infrared and visible imagery for object tracking: Toward trackers with superior IR performance*, 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 1–9.
5. F. XU, X. LIU, K. FUJIMURA, *Pedestrian detection and tracking with night vision*, IEEE Transactions on Intelligent Transportation Systems, **6**, *1*, pp. 63–71, 2005.
6. X. WANG, Z. TANG, *Modified particle filter-based infrared pedestrian tracking*, Infrared Physics & Technology, **53**, *4*, pp. 280–287, 2010.
7. C.N. PADOLE, L.A. ALEXANDRE, *Motion based particle filter for human tracking with thermal imaging*, 2010 3rd International Conference on Emerging Trends in Engineering and Technology, 2010, pp. 158–162.
8. Z. CUI, J. YANG, S. JIANG, J. LI, Y. GU, *Robust spatio-temporal context for infrared target tracking*, Infrared Physics & Technology, **91**, pp. 263–277, 2018.
9. L. CHEN, Z. LIU, *A robust and scalable method for infrared target identification*, Procedia Computer Science, **147**, pp. 172–176, 2019.
10. A. KRIZHEVSKY, I. SUTSKEVER, G.E. HINTON, *ImageNet classification with deep convolutional neural networks*, Communications of the ACM, **60**, *6*, pp. 84–90, 2017.
11. K. SIMONYAN, A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, International Conference on Learning Representations, 2015.

12. K. HE, X. ZHANG, S. REN, J. SUN. *Deep residual learning for image recognition*, IEEE Conference on Computer Vsion and Pattern Recognition, 2016, pp. 770–778.
13. L. BERTINETTO, J. VALMADRE, J.F. HENRIQUES, A. VEDALDI, P.H.S. TORR, *Fully-convolutional siamese networks for object tracking*, European Conference on Computer Vision, pp. 850–865, 2016.
14. C. MA, J.-B. HUANG, X. YANG, M.-H. YANG, *Hierarchical convolutional features for visual tracking*, IEEE international Conference on Computer Vision, 2015, pp. 3074–3082.
15. H. LI, Y. LI, F. PORIKLI, *Deeptrack: Learning discriminative feature representations online for robust visual tracking*, IEEE Transactions on Image Processing, **25**, pp. 1834–1848, 2015.
16. Y. QI, S. ZHANG, L. QIN, H. YAO, Q. HUANG, J. LIM, M.H. YANG, *Hedged deep tracking*, IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4303–4311.
17. J. VALMADRE, L. BERTINETTO, J. HENRIQUES, A. VEDALDI, P.H.S. TORR, *End-to-end representation learning for correlation filter based tracking*, IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2805–2813.
18. B. LI, J. YAN, W. WU, Z. ZHU, X. HU, *High performance visual tracking with siamese region proposal network*, IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8971–8980.
19. L. ENHAN, Z. RUI, Z. SHUO, W. RU, *An infrared pedestrian target tracking method based on video prediction*, J. Harbin Inst. Technol., **52**, pp. 192–200, 2020.
20. L. ZHENG, S. ZHAO, Y. ZHANG, L. YU, *Thermal infrared pedestrian tracking using joint siamese network and exemplar prediction model*, Pattern Recognition Letters, **140**, pp. 66–72, 2020.
21. D. YUAN, X. SHU, Q. LIU, Z. HE, *Structural target-aware model for thermal infrared tracking*, Neurocomputing, **491**, pp. 44–56, 2022.
22. W. LI, L. LV, J. ZHU, *Multigroup spatial shift models for thermal infrared tracking*, Knowledge-Based Systems, **255**, art. 109705, 2022.
23. Q. LIU, X. LU, Z. HE, C. ZHANG, W.S. CHEN, *Deep convolutional neural networks for thermal infrared object tracking*, Knowledge-Based Systems, **134**, pp. 189–198, 2017.
24. Y. SUN, X. WANG, X. TANG, *Deep convolutional network cascade for facial point detection*, IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3476–3483.
25. P. FELZENSZWALB, D. MCALLESTER, D. RAMANAN, *A discriminatively trained, multiscale, deformable part model*, 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
26. L. SEVILLA-LARA, E. LEARNED-MILLER, *Distribution fields for tracking*, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1910–1917.
27. A. BERG, J. AHLBERG, M. FELSBERG, *Channel coded distribution field tracking for thermal infrared imagery*, IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 9–17.
28. D.S. BOLME, J.R. BEVERIDGE, B.A. DRAPER, Y.M. LUI, *Visual object tracking using adaptive correlation filters*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544–2550.
29. J.F. HENRIQUES, R. CASEIRO, P. MARTINS, J. BATISTA, *Exploiting the circulant structure of tracking-by-detection with kernels*, 12th European Conference on Computer Vision, 2012, pp. 702–715.
30. J.F. HENRIQUES, R. CASEIRO, P. MARTINS, J. BATISTA, *High-speed tracking with kernelized correlation filters*, IEEE Transactions on Pattern Analysis Machine Intelligence, **37**, pp. 583–596, 2014.
31. T. LIU, G. WANG, Q. YANG, *Real-time part-based visual tracking via adaptive correlation filters*, IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4902–4912.
32. V. NARESH BODDETI, T. KANADE, B.V.K. VIJAYA KUMAR, *Correlation filters for object alignment*, IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2291–2298.
33. Q. LIU, Z. HE, X. LI, Y. ZHENG, *PTB-TIR: A thermal infrared pedestrian tracking benchmark*, IEEE Transactions on Multimedia, **22**, pp. 666–675, 2019.
34. T. XU, Z.-H. FENG, X.-J. WU, J. KITTLER, *Joint group feature selection and discriminative filter learning for robust visual object tracking*, IEEE/CVF International Conference on Computer Vision, 2019, pp. 7950–7960.
35. M. DANELLJAN, G. HÄGER, F. KHAN, M. FELSBERG, *Accurate scale estimation for robust visual tracking*, British Machine Vision Conference, 2014.
36. M. DANELLJAN, G. HAGER, F. SHAHBAZ KHAN, M. FELSBERG, *Learning spatially regularized correlation filters for visual tracking*, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4310–4318.
37. Q. LIU, X. LI, Z. HE, N. FAN, D. YUAN, H. WANG, *Learning deep multi-level similarity for thermal infrared object tracking*, IEEE Transactions on Multimedia, **23**, pp. 2114–2126, 2020.
38. X. LI, Q. LIU, N. FAN, Z. HE, H. WANG, *Hierarchical spatial-aware siamese network for thermal infrared object tracking*, Knowledge-Based Systems, **166**, pp. 71–81, 2019.
39. X. DONG, J. SHEN, *Triplet loss in siamese network for object tracking*, Proceedings of the European conference on computer vision (ECCV), 2018, pp. 459–474.
40. N. WANG, J. SHI, D.-Y. YEUNG, J. JIA, *Understanding and diagnosing visual tracking systems*, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3101–3109.
41. C. BAO, Y. WU, H. LING, H. JI, *Real time robust L1 tracker using accelerated proximal gradient approach*, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1830–1837.

Received July 4, 2022