Signal, Image and Video Processing (2022) 16:119–126
https://doi.org/10.1007/s11760-021-01968-5

ORIGINAL PAPER

Visual object tracking using Fourier domain phase information

Serdar Cakir[1] · A. Enis Cetin[2]

Received: 11 April 2021 / Revised: 16 May 2021 / Accepted: 16 June 2021 / Published online: 2 July 2021

Abstract
In this article, phase of the Fourier transform (FT), which has observed to be a crucial component in image representation, is utilized for visual target tracking. The main aim of the proposed scheme is to reduce the computational complexity of cross-correlation-based matching frameworks. Normalized cross-correlation (NCC) function-based object tracker is converted to a phase minimization problem under the following assumption: In visual object tracking applications, if the frame rate is high, the moving object can be considered to have translational shifts in image domain in a small time window. Since the proposed tracking framework works in the Fourier domain, the translational shifts in the image space are converted to phase variations in the Fourier domain due to the "translational invariance" property of the FT. The proposed algorithm estimates the spatial target position based on the phase information of the target region. The proposed framework uses the ℓ_1-norm and provides a computationally efficient solution for the tracking problem. Experimental studies indicate that the proposed phase-based technique obtain comparable results with baseline tracking algorithms which are computationally more complex.

Keywords Image phase information · Visual target tracking · Phase spectrum · Fourier transform · ℓ_1-norm

1 Related work on visual target tracking techniques

Visual object tracking is an important research area in the field of computer vision [1] which is utilized for various applications including surveillance [2], laser designation [3], transportation safety [4], human–computer interaction [5], and medical analysis [6]. Visual tracking problem is generally defined as the estimation of the target location in the image given some initial conditions such as the initial position and size of the target. During target tracking, position estimation may become difficult due to occlusions by other objects, object shape deformations, motion blur caused by rapid movements, illumination variations, low contrast between foreground and background, and changes in object scale [1].

✉ Serdar Cakir
cakir@bilkent.edu.tr

A. Enis Cetin
aecyy@uic.edu

[1] Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey

[2] Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA

In the literature, feature-based tracking frameworks have been utilized in order to find a representative feature set which may provide an increase in tracker performance. Depending on visual and spectral characteristics of the target, the selection of features may change. In the literature, physical properties of target such as color, edges, textures, etc., have been widely used for target representation in target tracking [7]. Direct pixel and pixel statistics-based approaches are also utilized for target tracking [8]. In addition, kernel-based techniques are preferred in challenging tracking applications [9–11]. As a strong feature descriptor for target representation, scale invariant feature transform (SIFT) [12] has also been used in target tracking and target classification [13,14]. Covariance feature descriptor [15] which is computationally more efficient than SIFT-based approaches, has been also utilized for target tracking applications [16–18]. Although feature descriptors and complex tracking frameworks [19] have been very successful in tracking applications, the computational complexity of the solution is important factor for an efficient implementation. Unfortunately, the complex descriptor-based techniques may not be implemented in real time. However, tracking objects in real time is crucial in visual surveillance applications.

Template matching-based approaches have been used in many fields due to their easy implementation and low com-

putational cost [20]. The idea behind these approaches is to measure the similarity between two image or feature patches using various functions to quantify the matching quality. The NCC function has been widely used for template matching [21,22], and it can be utilized in both spatial and frequency domain [23]. However, when the template size becomes large, the computational complexity of the NCC function grows geometrically. In order to reduce the computational cost of the NCC algorithm, some researches developed efficient implementations [23–25].

In this article, we propose a computationally efficient visual target tracker scheme based on image phase information to reduce computational complexity of the template matching while preserving the tracking performance at an appropriate level. The proposed phase-based tracking scheme is described in Sect. 2. The experimental results and observations are presented in Sect. 3.

2 Phase based visual target tracking framework

Matching-based tracking techniques are generally triggered by a target detection mechanism which provides a target model. The detector framework may be implemented manually or automatically depending on the availability of an operator. After the target region is obtained by the detector, the matching-based tracking scheme tries to match similar regions to the target model during tracking process. While searching for the new target location, the target tracker generally performs a limited search around the previous target location to satisfy computational limitations. In this way, instead of processing the whole image, it performs the search within a sub-window that is generally called as "search region". Performing target search in search region not only decreases the computational complexity but also increases the tracking performance by filtering out the objects which may have similar appearance to the target of interest. In order to quantify the similarity between search region patches and the target model, different similarity measures such as sum of squared differences (SSD), sum of absolute differences (SAD), and normalized cross-correlation (NCC) have been used [26].

The NCC matching function is one of the most popular matching schemes in the literature due to its efficiency and applicability to a wide variety of applications. In general, NCC can be calculated as follows:

$$\gamma[p,q] = \frac{\sum_{x,y}\left[s[x,y]-\bar{s}_{p,q}\right]\left[t[x-p,y-q]-\bar{t}\,\right]}{\left\{\sum_{x,y}\left[s[x,y]-\bar{s}_{p,q}\right]^2\sum_{x,y}\left[t[x-p,y-q]-\bar{t}\,\right]^2\right\}^{1/2}} \tag{1}$$

where s is the search image, t is the target template, $\bar{t}$ is mean value of the target template, and $\bar{s}_{p,q}$ is the mean value of $s[x,y]$ which is the candidate target region in the search window. In Eq. 1, the computational complexity of the numerator is significantly high although some researchers utilize transform domain approaches to reduce computational complexity [23]. Moreover, the computational time of the NCC algorithm increases when the target region becomes large. To overcome these limitations, our aim is to propose a computationally efficient framework for template matching problem. The proposed technique makes use of image phase information to perform template matching. The proposed scheme also relies on the following assumption: Most of the moving objects result in translational shifts in the image domain in a small time window. This assumption is valid when the frame rate is high. Since the proposed tracking framework works in the Fourier domain, the translational shifts in the image space are converted to phase variations in the Fourier domain due to the "translational invariance" property of the FT. The proposed technique is derived by using general definition of NCC formulation. The nominator of NCC function (Eq. 1) is the cross-correlation between the target template and search region. By making use of the "cross-correlation theorem", the frequency domain equivalent of the cross-correlation process is calculated as follows:

$$\Gamma[u,v] = S_{p,q}[u,v]T^*[u,v] \tag{2}$$

where $S_{p,q}[u,v]$ is the Fourier transform of the sub-region in the search region and $T^*[u,v]$ is the complex conjugate of the Fourier transform of the target model. $\Gamma[u,v]$ in Eq. 2 can be rewritten as follows:

$$\Gamma[u,v] = \left|S_{p,q}[u,v]\right| e^{j\phi_{p,q}[u,v]}\,|T[u,v]|\,e^{-j\phi_T[u,v]} \tag{3}$$

where $\phi_{p,q}[u,v]$ and $\phi_T[u,v]$ are the phase of candidate target region and target template, respectively.

Recalling that the moving objects in the scene result in translational shifts in a small time frame, magnitudes of the Fourier transforms of the candidate region $\left|S_{p,q}[u,v]\right|$ and target model $|T[u,v]|$ are approximately the same. In other words, using only the phase information between the candidate region and target model may be sufficient for the target tracking problem. In the ideal case, the target model and candidate region are perfectly matched and the phase difference between these regions becomes zero. Therefore, the template matching problem can be rewritten as the following ℓ_1 minimization problem:

$$\min \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|\phi_{p,q}[u,v]-\phi_T[u,v]\right| \tag{4}$$

where M, N denote the size of 2D discrete Fourier transform (DFT). By looking at the calculations presented in Eq. 4, one can see that the minimization problem does not require any multiplications on the DFT grid. Therefore, the matching process can be performed efficiently. Moreover, there is no need to calculate the FFT over and over again for each image frame by using an approach derived in Eq. 5. When there is a translational shift in the image space, it corresponds to phase variations in the Fourier domain due to the translational invariance property of the FT. The phase variations corresponding to each translational shift can be related as follows:

$$\phi_{shift}[u,v] = \phi_T[u,v] - 2\pi\left(\frac{um}{M} + \frac{vn}{N}\right) \tag{5}$$

The second term in Eq. 5 can be stored in a look-up table to achieve computational savings. In other words, artificial translational shifts can be generated based on the target model t, and phase variations corresponding to each artificial translational shift can be stored in memory. This operation is carried out once after each target detection. In this way, the repetitive FFT calculation in each image frame is no longer necessary for the matching problem and a computationally efficient solution for the target tracking problem can be achieved.

In order to construct look-up tables for phase variations, artificial translational shifts can be performed as follows:

$$t_{(m,n)}[x,y] = t[x-m, y-n] \tag{6}$$

where m and n are integers in $[-\epsilon, \epsilon]$ interval. The phase variations $\phi_{T_{(m,n)}(u,v)}$ corresponding to each artificial translational shift can be obtained by computing the phase of the Fourier transform of $t_{(m,n)}$. Then, the target region in the current frame can be obtained by redefining the minimization problem in Eq. 4 as follows:

$$\phi_{T_{(m^*,n^*)}}[u,v] = \underset{m,n}{\text{minimize}} \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|\phi_C[u,v] - \phi_{T_{(m,n)}}[u,v]\right| \tag{7}$$
$$subject\ to\ m,n \in [-\varepsilon,\varepsilon]$$

where $\phi_C[u,v]$ is the phase information of the image region in the current frame where the target was located in the previous frame. The translational shift (m^*, n^*) providing the best match with the current target region $c(x,y)$ is determined as the offset between the current and previous target location. If the target of interest is located at (x_{t-1}, y_{t-1}) in the previous video frame, the new target location (x_t, y_t) can be obtained as follows:

$$\begin{aligned} x_t &= x_{t-1} + m^* \\ y_t &= y_{t-1} + n^* \end{aligned} \tag{8}$$

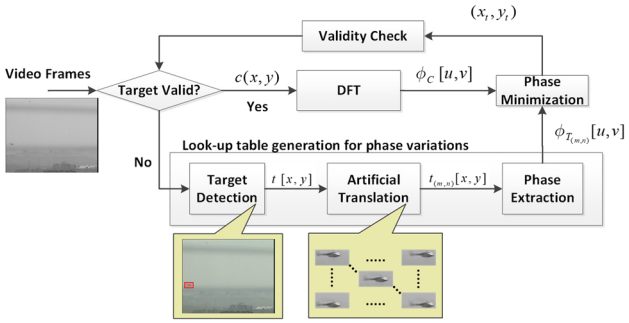


Fig. 1 Flow diagram of the proposed phase-based tracking scheme

Also note that there is no need to normalize the phase-based matching function described in Eq. 7. This is because the phase of the Fourier transform is not affected by the amplification factors which are all real valued.

The overall matching process is repeated for each video frame. The proposed tracker framework can be summarized by the flow diagram presented in Fig. 1.

In Fig. 1, the tracking framework is triggered by target detection which can be utilized manually or automatically. After the target region is determined, look-up tables for phase variations are generated by introducing artificial translational shifts to the target template. Here, look-up tables enable a computationally more efficient scheme for target tracking. By using these pre-stored look-up tables, the current target position is obtained by solving the phase minimization problem defined in Eq. 7. After the target region is obtained in the current frame, the validity of the target model is checked by simply comparing the resultant phase difference in Eq. 7. The target validation measure can also be selected from a large variety of functions from pixel-wise error measures to image quality metrics between the target template and candidate target region. If the target region obtained by the tracker is valid, the tracking loop continues to operate in tracking mode. Otherwise, the tracking scheme switches to "reacquisition" mode. In reacquisition mode, the detection block tries to determine the target location.

3 Experimental studies

The performance of the proposed tracking scheme is tested using several image sequences containing sea-surface targets, aerial targets and ground vehicles in an urban environment. In order to measure the performance of the proposed tracker, the target region extracted by the tracker is compared with the ground-truth information (actual location of the target on each video frame). In order to explain and determine the performance measures visually, the bounding rectangles obtained by the tracker and ground-truth information are illustrated in Fig. 2.

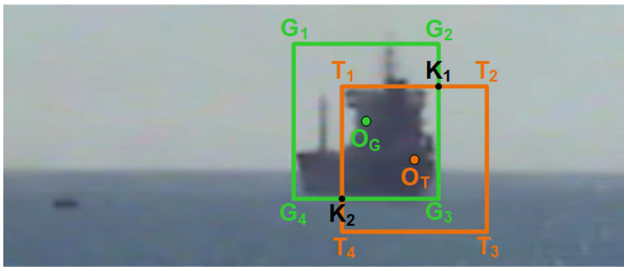


Fig. 2 The illustration of bounding rectangles obtained by the tracker and data ground-truth information. Rectangle $G_1G_2G_3G_4$: actual target gate obtained by ground-truth information. Rectangle $T_1T_2T_3T_4$: example target gate obtained by a tracking algorithm

In order to quantify the tracking performance, objective measures described in [17–19] are used. These objective measures compare the bounding boxes around the target region determined by the tracker and ground-truth information. We also make use of a commonly used tracking benchmark [19] in order to constitute a fair and more up-to-date performance evaluation. The tracker performance is evaluated using the success and precision rates defined in [19].

The first performance measure, the success rate, is based on the overlapping region between the bounding boxes obtained by the tracker and ground-truth information. After the computation of overlapping region, the "overlapping score β" is obtained by normalizing the overlapping region with the union of the target bounding boxes obtained by the tracker and ground-truth information. By using the illustration presented in Fig. 2, the overlapping score can be defined as follows:

$$\beta = \frac{\text{Area}\,(T_1K_1G_3K_2)}{\text{Area}\,(G_1G_2K_1T_2T_3T_4K_2G_4)} \tag{9}$$

The overlapping score β varies between $(0, 1)$ depending on the bounding boxes produced by tracker and ground-truth information. If the target bounding box obtained by the tracker is exactly the same as the actual target bounding box, β score becomes 1 which is a sign of exact overlap. The overlapping score β is calculated for each frame of the image sequence. In order to generate success plots [19], a threshold τ_S is determined to control the level of overlapping. Then, the number of frames which provide an overlapping score β larger than τ_S is counted. This process is repeated for each τ_S value in the $(0, 1)$ interval. At the end, the success rate is computed as the ratio of frames which provide β values which are larger than the τ_S. Since the success rate determines the relation between the overlapping frame ratio and the overlapping threshold, it is generally called as "success plot".

The second performance measure, precision rate, is based on the pixel-wise distance between the center coordinates of the bounding boxes (O_G and O_T in Fig. 2) obtained by the tracker and ground-truth information. The precision rate calculation starts by calculating the Euclidean distance between the centers of the target gates produced by the tracker and ground-truth information for each frame of the video. In order to generate precision plots, a threshold τ_P is determined to control the level of pixel-wise error between the centers of the bounding boxes. Then, the number of frames that provides a pixel-wise error smaller than the error threshold τ_P is counted. This process is repeated for each τ_P value varying between $(0, \xi)$ interval ($\xi = 50$ in our experiments). At the end, the precision rate is computed as the ratio of frames which provide pixel-wise error values smaller than the τ_P which varies between $(0, 1)$. The precision rate is also named as "precision plot".

In addition to success and precision rates, three ranking measures, namely area under curve (AUC), track maintenance (TM), and localization accuracy (LA), are used in order to rank the trackers according to their performance. The ranking measures have been widely utilized to quantify the overall performance evaluation of the trackers [19,27]. The AUC and TM measures are extracted from the success rate, while the LA measure is derived from the precision plot. The AUC measure corresponds to the area under the success plot, and TM measure is the percentage of frames in which a nonzero overlap occurs between the target gates obtained by the tracker and ground-truth information. The LA measure is defined as the percentage of frames in which the pixel-wise error in localization of the target by tracker is below a certain threshold. In this work, the acceptable pixel-wise error is selected as 10 pixels. Therefore, the LA measure is determined as the precision value corresponding to 10-pixel error threshold.

3.1 Dataset

In the experiments, a database is constructed by using six videos containing moving vehicles in outdoor environments. After capturing the videos, targets which are intended to be tracked are determined. In order to create a controlled test set, the ground-truth information is extracted by an expert, manually. Here, the operator determines the target rectangle manually at certain frames using a customized software. The videos used in the experiments are captured by a visible band and a long-wave infrared (LWIR) camera. The first video ("Video_Seq_1") contains 1000 frames of a moving sea-surface platform that is occluded by other sea-surface targets in certain frames. Video_Seq_1 contains a good scenario to test the robustness of the tracking algorithm when there is a partial occlusion on the target. The second video ("Video_Seq_2") contains 500 frames of a moving fishing boat on a complex background. The back-

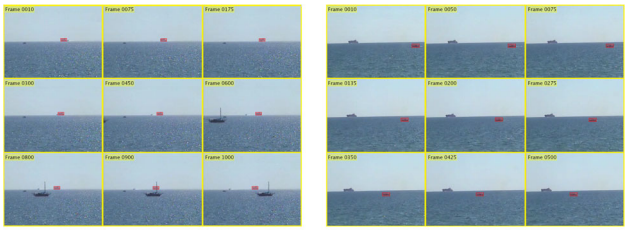
Table 1 The basic properties of the videos used in the performance evaluation

Video/Scenario properties					
Name	Type	Image Size	# of frames	Scenario	Platform
Video_Seq_1	Visible	640 × 480	1000	Nautical	Ship
Video_Seq_2	Visible	640 × 480	500	Nautical	Ship
Video_Seq_3	Visible	720 × 576	120	Aerial	Helicopter
Video_Seq_4	Visible	640 × 480	200	Ground	Motorcycle
Video_Seq_5	LWIR	320 × 240	198	Ground	Car
Video_Seq_6	LWIR	640 × 480	625	Nautical	Ship

ground is complex due to sea glints caused by the reflection of the sunlight on the waves. The third video sequence ("Video_Seq_3") contains a considerably fast moving aerial platform. The aerial platform is exposed to illumination changes in certain frames of the total 120 video frames. Moreover, the visibility is low in the captured frames due to the atmospheric conditions of the time of recording. The fourth video ("Video_Seq_4") contains 200 frames of a moving motorcycle in an urban environment. Since the video is captured in a populated area, the background contains other moving objects. Moreover, the capturing device is not stabilized, and some of the video frames are blurred due to undesired movements of the capturing device. The fifth video ("Video_Seq_5") contains 198 frames of a moving car in an urban environment. The Video_Seq_5 is captured by a LWIR camera, and the video contains illumination changes and partial occlusions in certain frames. The last video (Video_Seq_6) used in the experiment contains 625 frames of an approaching ship captured by an unstabilized infrared camera which is exposed to undesired vibrations throughout the video. This video, which is originally named as "boat1", is obtained from the Visual Object Tracking dataset (VOT-TIR2016) [28,29]. Table 1 provides a brief summary about the videos used in the experiments.

3.2 Baseline techniques

The proposed phase tracker is compared with the baseline techniques using the objective performance measures over the six videos. The baseline techniques used for the comparison are: The Discriminative Scale Space Tracker (DSST) [30], Fast Compressive Tracker (FCT) [31], Incremental Visual Tracker (IVT) [32], kernelized correlation filter (KCF) [33], sum of template and pixel-wise learners (Staple) tracker [34], Minimum Output Sum of Squared Errors (MOSSE) tracker [35], SRDCF [36], and NCC [23]. These techniques are well established and sophisticated tracking schemes which had participated in tracking benchmarks.



(a) Video_Seq_1 **(b)** Video_Seq_2

Fig. 3 Sample target gates produced by the proposed tracking scheme in Video_Seq_1 and Video_Seq_2

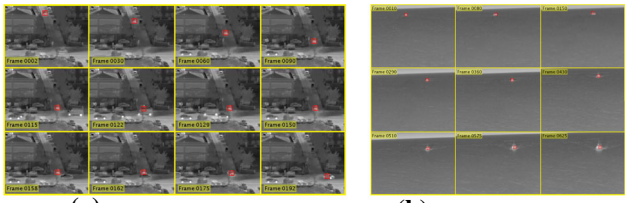
(a) Video_Seq_3 **(b)** Video_Seq_4

Fig. 4 Sample target gates produced by the proposed tracking scheme in Video_Seq_3 and Video_Seq_4

3.3 Tracking experiments on different scenarios

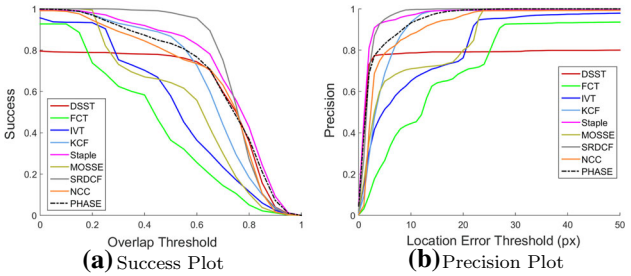
The proposed phase tracker is tested by using all of the six videos in the dataset. The target gates produced by the tracker are stored at each frame to compare the actual target gates obtained with the ground-truth information. To visualize the tracking process, the target gates produced by the proposed tracking scheme are marked with red symbologies for each video sequence. Also, each video frame processed by the tracker is marked with the frame information to make the tracking scenario practical to follow. Example target gates for each video are presented in Figs. 3a, b, 4a, b, 5a, b, respectively.

In early frames of Video_Seq_1, the target is occluded by a moving speed boat that produces sea glints caused by the motor. Although the appearance of the target changes significantly, the phase-based tracker has been able to track the target. In the last frames of the same video sequence, the occlusion source (sailboat) has more contrast than the tar-



(a) Video _Seq_5 **(b)** Video _Seq _6

Fig. 5 Sample target gates produced by the proposed tracking scheme in Video_Seq_5 and Video_Seq_6



(a) Success Plot **(b)** Precision Plot

Fig. 6 The success and precision plot of the overall performance evaluation

get of interest. In other words, the clutter is more significant than the original target. However, phase tracker maintains the lock on the desired target. The Video_Seq_2 contains a complex background which changes rapidly due to sea-glints. However, the proposed phase-based tracker is able to track the target successfully throughout the frames. In the Video_Seq_3, the target model changes rapidly due to illumination variations and motion blur in certain frames. In spite of these difficulties, the proposed tracking scheme does not lose the lock on the target throughout the video. Captured in an urban environment, Video_Seq_4 has a complex background which contains moving vehicles near the target of interest. Also, some of the frames in Video_Seq_4 are blurry due to the undesired rapid movements of the capturing device. Although the proposed scheme is able to track the target throughout the video, there are noticeable localization errors in some frames which cause performance loses in terms of objective metrics. The Video_Seq_5 is an infrared image sequence captured in an urban environment. In this video, there are significant illumination changes in certain frames and partial occlusions. Moreover, the background of the video is complex due to different heat sources near the target of interest. In the last frames of the video, the target model is even exposed to sudden rotation changes caused by the maneuver of the target. The proposed tracker is able to track the target throughout these frames, but noticeable tracking errors occur in some frames. Most of the frames in Video_Seq_6 are effected by Gaussian blur due to the undesired movements of the capturing device. Additionally, the scale of the target changes gradually since the target is approaching the capturing device during the video capture. Despite these difficulties, the proposed phase-based scheme is able to maintain the track on the target. However, the scale changes of the target result in degradation in tracking performance of the proposed phase tracker.

We expand the experimental studies to compare the performance of the proposed tracking scheme with baseline tracking techniques. By using the target gates obtained for each tracker, the performance measures, success rate and precision rate, are computed for each of the videos in the dataset. The overall performance plots for all six videos are presented in Fig. 6.

The results presented in Figs. 3a, 6 show that the proposed tracking technique provides a consistent regime on performance test. In the overall evaluation of the video sequences, the proposed phase tracker seems to track the target with acceptable success and precision rates. Since the video sequences contain negative factors such as illumination variations, scale changes, motion blur, complex background and partial occlusions, the results provided by the proposed technique are quite satisfactory. Developed with the aim of reducing the computational complexity of the NCC algorithm, the proposed phase tracker obtains slightly better results than the NCC tracker. It is not surprising that some of the baseline tracking algorithms, which are sophisticated and computationally complex tracker frameworks, outperform the proposed tracking scheme in objective performance evaluations. Although it is simple by design, the proposed phase tracker achieves satisfactory tracking accuracy while outperforming NCC, DSST, FCT, IVT, and MOSSE trackers in the tests. Similar to the proposed phase-based tracking framework, the NCC tracker obtains satisfactory results on all of the scenarios.

In order to summarize the performance tests of the trackers, the overall performance of each tracker is evaluated by computing the ranking measures (AUC, TM, and LA) over

Table 2 The overall performance evaluations of video trackers

	Ranking method		
	Success		Precision
	AUC	TM	LA
DSST [30]	0.6055	0.7951	0.7860
FCT [31]	0.4186	0.9267	0.4439
IVT [32]	0.5138	0.9561	0.6507
KCF [33]	0.6545	0.9936	0.9350
Staple [34]	0.7235	0.9936	0.9694
MOSSE [35]	0.5476	1.0000	0.7093
SRDCF [36]	0.7487	1.0000	0.9958
NCC [23]	0.6639	0.9909	0.8760
Proposed	0.6886	0.9989	0.9316

Table 3 Average time required to process a single frame by the proposed and baseline tracking algorithms

	Computation time versus Target size (Pixel)				
	$32 \times 32(ms)$	64×64 (ms)	128×128 (ms)	256×256 (ms)	512×512 (ms)
DSST [30]	34.8	63.4	180	685	2480
FCT [31]	8.4	8.1	8.2	8.4	8.7
IVT [32]	18.9	18.3	19.7	18.8	18.5
KCF [33]	3.1	4.9	6.8	23.4	87.2
Staple [34]	10	10.6	16.7	45.6	213
MOSSE [35]	3.3	6.9	21.4	93.7	397
SRDCF [36]	67.5	152	177	270	612
NCC [23]	0.9	2.5	8	38.6	120
Proposed	0.4	0.7	2.7	7.5	18.7

the frames of all videos. The results obtained by the overall performance evaluations are presented in Table 2.

The overall performance evaluations presented in Table 2 show that the proposed phase tracker provides comparable results with the baseline tracking frameworks. The proposed scheme obtains a satisfactory AUC result which is a sign of decent localization of the target gate throughout the frames. Moreover, the phase tracker is able to track the target successfully by producing very high TM score which corresponds to very few non-overlapping states of the target gates produced by the tracker and ground-truth information. In the experiments, although the LA is evaluated at a tight error threshold of 10-pixels, the localization error of the target locations obtained by the proposed tracker is below 10 pixels in approximately 93% of the all video frames.

As an additional experiment, the proposed tracking scheme is compared with the NCC tracker as well as other baseline techniques in terms of computational complexity. The main aim of this experiment is to evaluate and compare the computation times of the proposed tracker and baseline techniques with respect to varying dimensions of the target model. The system specification should be taken into account when discussing computation time. The proposed tracker framework is implemented in Matlab environment on a computer containing Intel(R) Core(TM) i5-10400F 2.90GHz processor, 16 GB RAM running on Microsoft Windows 10 operating system. Starting from a target model of size 32×32, the width and height of the target model are doubled at each step until we obtain a target model of size 512×512. At each step of the experiment, the average time required to process a single frame by the trackers is computed. The results are presented in Table 3.

The results presented in Table 3 indicate that the proposed phase tracker is computationally more efficient compared to the classical NCC framework as well as other baseline techniques. Even for the largest target size (512×512), the proposed scheme performs within the limit of real-time requirements. Moreover, the processing time of the proposed phase tracker does not face an rapid growth with the increasing values of target size.

4 Conclusion

In this paper, image phase information is used to reduce the computational complexity of template matching-based tracking frameworks. The proposed phase-based tracker is able to reduce the computational complexity while preserving the tracking performance at an appropriate level. The proposed scheme can be an alternative to the NCC-based trackers due its low computational load and fast response. In terms of tracking performance, the proposed framework has obtained comparable results with the DSST [30], KCF [33], Staple [34], SRDCF [36], and NCC [23] trackers while outperforming more complicated tracking frameworks such as FCT [31], IVT [32], and MOSSE [35].

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. **38**(4), 13 (2006)
2. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE **90**(7), 1151–1163 (2002)
3. Clendenin, R., Freeman, R.: Optical target tracking and designating system, US Patent 4,386,848 (1983). https://www.google.com/patents/US4386848
4. Smith, C.E., Richards, C.A., Brandt, S.A., Papanikolopoulos, N.P.: Visual tracking for intelligent vehicle-highway systems. IEEE Trans. Veh. Technol. **45**(4), 744–759 (1996)
5. Jacob, R., Karn, K.S.: Eye tracking in human-computer interaction and usability research: ready to deliver the promises. Mind **2**(3), 4 (2003)
6. Günther, J., Bongers, A.: 3d motion detection and correction by object tracking in ultrasound images, US Patent 8,348,846 (2013). https://www.google.com/patents/US8348846

7. Deori, B., Thounaojam, D.M.: A survey on moving object tracking in video. Int. J. Inf. Theory **3**(3), 31–46 (2014)
8. Nishida, K., Kurita, T., Ogiuchi, Y., Higashikubo, M.: Visual tracking algorithm using pixel-pair feature. In: International Conference on Pattern Recognition, pp. 1808–1811 (2010)
9. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
10. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662), Vol. 2, pp. 142–149 (2000)
11. Bradski, G.R.: Computer vision face tracking for use in a perceptual user interface (1998)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
13. Lee, H., Heo, P.G., Suk, J.-Y., Yeou, B.-Y., Park, H.: Scale-invariant object tracking method using strong corners in the scale domain. Opt. Eng. **48**(1), 7204 (2009)
14. Park, C., Baea, K.-H., Jung, J.-H.: Object recognition in infrared image sequences using scale invariant feature transform. Proc. SPIE **6968**, 69681 (2008)
15. Tuzel, O., Porikli, F., Meer, P.: Region covariance: a fast descriptor for detection and classification. Comput. Vis. ECCV **2006**, 589–600 (2006)
16. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on lie algebra. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, IEEE, pp. 728–735 (2006)
17. Cakır, S., Aytaç, T., Yıldırım, A., Gerek, Ö.N.: Classifier-based offline feature selection and evaluation for visual tracking of sea-surface and aerial targets. Opt. Eng. **50**(10), 107205–107205 (2011)
18. Cakir, S., Aytaç, T., Yildirim, A., Beheshti, S., Gerek, Ö.N., Cetin, A.E.: Salient point region covariance descriptor for target tracking. Opt. Eng. **52**(2), 027207–027207 (2013)
19. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: a benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2411–2418 (2013)
20. Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. Wiley, New York (2009)
21. Briechle, K., Hanebeck, U.D.: Template matching using fast normalized cross correlation. Science **4387**, 95–102 (2001)
22. Goshtasby, A., Gage, S.H., Bartholic, J.F.: A two-stage cross correlation approach to template matching. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **6**(3), 374–378 (1984)
23. Lewis, J.P.: Fast normalized cross-correlation. Vis. Interface **10**, 120–123 (1995)
24. Luo, J., Konofagou, E.E.: A fast normalized cross-correlation calculation method for motion estimation. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **57**(6), 1347–1357 (2010)
25. Fouda, Y., Ragab, K.: An efficient implementation of normalized cross-correlation image matching based on pyramid. In: 2013 International Joint Conference on Awareness Science and Technology Ubi-Media Computing (iCAST 2013 UMEDIA 2013), pp. 98–103 (2013)
26. Ouyang, W., Tombari, F., Mattoccia, S., Stefano, L.D., Cham, W.K.: Performance evaluation of full search equivalent pattern matching algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **34**(1), 127–143 (2012)
27. Demir, H.S., Cetin, A.E.: Co-difference based object tracking algorithm for infrared videos. IEEE Int. Conf. Image Process. **2016**, 434–438 (2016)
28. Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2137–2155 (2016)
29. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojir, T., Häger, G., Lukežič, A., Fernandez, G.: http://www.springer.com/gp/book/9783319488806The visual object tracking vot2016 challenge results, Springer (2016). http://www.springer.com/gp/book/9783319488806
30. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, BMVA Press (2014)
31. Zhang, K., Zhang, L., Yang, M.H.: Fast compressive tracking. IEEE Trans. Pattern Anal. Mach. Intell. **36**(10), 2002–2015 (2014)
32. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1), 125–141 (2008)
33. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
34. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: Complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409 (2016)
35. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. **2010**, 2544–2550 (2010)
36. Danelljan, M., Hager, G., Shahbaz, K.F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)