Transactions on Machine Intelligence 7(4) (2024) 257-268

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Video Tracking Using Correlation Measurement

A. Toobaei [1,*], F. Mehna [2]

[1] Department of Electrical and Telecommunications Engineering, Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran
[2] Assistant Professor, Department of Electrical and Telecommunications Engineering, University of Sistan and Baluchestan, Zahedan, Iran

ARTICLE INFO

Article History:
Received 6 April 2024
Received in revised form 21 July 2024
Accepted 29 November 2024
Available online 4 December 2024

Keywords:
Video Tracking, Motion Model, Observation Model, Correlation

ABSTRACT

Video tracking is one of the fundamental and widely used topics in the field of computer vision, with applications ranging from surveillance and security systems to autonomous vehicles and human-computer interaction. Despite extensive research and significant advancements in this area, numerous challenges continue to affect the performance and reliability of visual tracking systems. Among these challenges are occlusion, where objects are partially or completely hidden from view; lighting changes, which can alter the appearance of objects; scale and size variations; rapid object movements; background clutter; and computational complexity, which hinders real-time processing capabilities. In this paper, a particle filter is employed as the motion model to predict and estimate the states of the target object over time. The observation model evaluates the likelihood of the predicted states based on their correlation with a predefined pattern or template. By leveraging correlation measurement for state evaluation, the proposed video tracking method can address several of the common challenges, such as changes in lighting and partial occlusion. Moreover, because correlation measurement typically requires lower computational resources compared to more complex tracking algorithms like deep learning-based methods, the proposed approach enables faster processing speeds. Consequently, it offers improved performance for real-time applications where both accuracy and efficiency are critical.

1. INTRODUCTION

Video tracking is one of the topics widely used in the field of computer vision, and much research has been conducted in this area [1-3]. Some of the applications of target tracking include motion analysis, intelligent surveillance, traffic control, military applications, and more [5,4,2]. The task of target tracking is to estimate and locate the position and different states of an object within a video sequence [6,1]. Despite significant advancements in visual tracking, there are still challenges and issues that make designing a robust algorithm difficult [7,3]. Some

* Corresponding Author: a.toobaei.110@gmail.com
Department of Electrical and Telecommunications Engineering, Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran

http://dx.doi.org/10.47176/TMI.2024.257

of these problems include occlusion, lighting changes, scale variation, sudden and rapid movement, blurring, computational complexity, and so on [2-8].

The motion model, which is responsible for estimating the object's states, and the observation (appearance) model, which displays the object and evaluates the probability of estimated states in the motion model, are the two key components in target tracking [6,4,2]. The combination of the motion model and the observation model must be optimized to ensure efficient visual tracking [2].

Generally, the observation model can be classified into three types: generative, discriminative, and hybrid models [2]. Generative methods usually train an appearance model to represent the target object and use it to find a region in the image that has the least reconstruction error or the greatest similarity to that model [8,2]. This approach is typically used when limited data is available [2]. Discriminative methods treat target tracking as a binary classification problem, defining a boundary to separate the target from the background [8,2]. These algorithms focus on distinguishing the target from its surrounding background using positive and negative samples [8]. This approach works better when the training set is large [2]. Hybrid methods are used to combine both approaches and exploit their advantages [2].

In this paper, the particle filter is used in the motion model. The particle generation in this filter is based on a random walk model [9] to estimate different states in the motion model. Additionally, correlation measurement is employed in the observation model, which can be categorized as a generative method. The particle filter used in the motion model helps optimize the combination of the motion and observation models.

2. RELATED WORK

Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene [11,10]. This means that a tracker continuously labels the object being tracked across different frames [10]. Object detection and how it moves are the first steps in the object tracking process. The object tracking process must be supported by additional methods to classify the detected objects [12]. Some of the most important object tracking methods include point tracking, kernel tracking, and shadow tracking.

2.1. Point Tracking Method

In point tracking, objects can be represented by points in consecutive frames, and the assignment of these points is based on the previous shape state, which may include the object's position and movement. This method requires an external mechanism to detect the object in each frame [13]. Point tracking methods can be divided into deterministic and probabilistic categories. Deterministic methods use qualitative motion prediction methods to solve the object similarity problem. Probabilistic methods use criteria such as object scale and size, and uncertainty to solve the object similarity issue [12]. Kalman and particle filters are prominent examples of these methods.

2.2. Kernel Tracking Method

Kernel tracking is typically performed by calculating the moving object, which is represented by an initial object region that moves from one frame to the next. The movement of the object is usually represented in the form of parametric motion models like translation, affine, etc. [11]. These algorithms differ in terms of the number of objects tracked and the methods used for approximating the object's movement [11].

In real-time applications, object representation using geometric shapes is common. However, one limitation is that some part of the object might fall outside the defined shape, while part of the background might be included inside the shape [13].

2.3. Shadow Method

The goal of this tracking method is to find the target object's region in each frame using a model of the object that is generated by the previous frame [10].

The model for tracking the object in each frame is created using previous frames. This method is capable of tracking various shapes of objects, solving the occlusion problem, tracking objects that are divided into smaller objects, and tracking objects formed by the merging of other objects [10].

Some objects, such as hands, fingers, shoulders, etc., have complex shapes that cannot easily be compared to simple geometric shapes. Shadow-based methods use a detailed shape description for the objects [14].

3. PROPOSED METHOD

The method presented in this paper is based on the use of a particle filter and a correlation criterion. In the following sections, we will explain more about the particle filter and the correlation criterion, as well as their application in the proposed algorithm.

3.1. Particle Filter

The particle filter is an important sequential sampling technique based on Bayes' theorem, which attempts to estimate the posterior distribution of state variables for a given dynamic system [2]. This filter uses a set of weighted particles to approximate the state probability distribution, regardless of its underlying distribution, making it highly effective for handling nonlinear and non-Gaussian systems [2]. As a typical state inference problem, online image registration can be modeled by the particle filter [15].

There are two main steps in the particle filter method: 1) prediction and 2) update [2]. In this sampling technique, based on how the observations were up to the previous frame (t-1), the probability of selecting different state variables in the current frame (t) is calculated, and the state variable in the current frame is predicted. Additionally, the filter uses the current state variable's probability based on observations up to the current frame to update itself. In other words, the prediction and update steps in the particle filter recursively estimate the next state probability based on the following two rules [2].

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})\, p(x_{t-1}|y_{1:t-1})\, dx_{t-1} \tag{1}$$

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)\, p(x_{t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{2}$$

The expression x_t represents the state variable in frame t, which describes the relative motion parameters of an object, and y_t is the observation vector corresponding to x_t. The relative motion parameters of an object are shown in equation (3), which, from left to right, represent translation in the x and y directions, rotation angle, scale, aspect ratio (the ratio of length to width of the tracking box), and skew.

$$x_t = \{x_t, \mathrm{y}_t, \theta_t, \mathrm{s}_t, \alpha_t, \phi_t\} \tag{3}$$

3.2. Correlation Criterion

The correlation coefficient is a measure used to assess the similarity between two variables, and its mathematical relationship is derived as follows.

$$corr(A, B) = \frac{cov(A,B)}{\sigma_A \sigma_B} \tag{4}$$

In Equation (4), the denominator of the fraction is equal to the product of the standard deviations of each variable, and the numerator is equal to the covariance of the two variables, calculated as follows.

$$cov(A, B) = E[(A - \mu_A)(B - \mu_B)] \tag{5}$$

Where E[.] denotes the expected value, and μ_A and μ_B are the means of variables A and B, respectively. Therefore, Equation (4) can be expressed as follows.

$$corr(A,B) = \frac{E[(A-\mu_A)(B-\mu_B)]}{\sqrt{E[(A-\mu_A)^2]}\sqrt{E[(B-\mu_B)^2]}} \tag{6}$$

If A and B are matrices, the correlation coefficient between them will be calculated as follows.

$$corr(A,B) = \frac{\sum_m \sum_n (A_{mn}-\mu_A)(B_{mn}-\mu_B)}{\sqrt{(\sum_m \sum_n (A_{mn}-\mu_A)^2)(\sum_m \sum_n (B_{mn}-\mu_B)^2)}} \tag{7}$$

Additionally, if A and B are vectors, this relationship is expressed as Equation (8).

$$corr(A,B) = \frac{\sum_k ((A_k-\mu_A)(B_k-\mu_B))}{\sqrt{(\sum_k (A_k-\mu_A)^2)(\sum_k (B_k-\mu_B)^2)}} \tag{8}$$

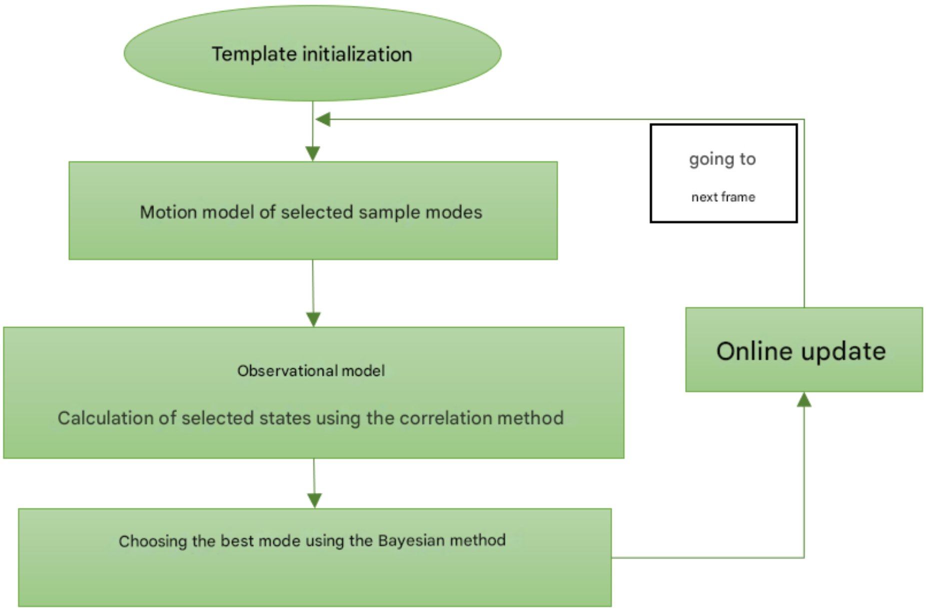


Fig. 1. Flowchart of the Correlation-Based Tracking Algorithm

3.3. Proposed Algorithm

In this paper, tracking is considered a particle filter process with a hidden Markov model. In this method, the initial position of the target is determined by specifying the center, width, and height of the tracking frame. This initial position is treated as a template and updated for each frame. The motion model predicts and estimates various states for tracking the target in the subsequent frame. Then, based on the observation model, the likelihood of each estimated state is evaluated, and the best sample from the predicted states is selected. Once the optimal sample and its observation model are obtained, the template is updated, and tracking continues for the following frames. Figure 1 illustrates the flowchart of the algorithm proposed in this paper.

3.3.1. Motion Model

To determine the best state from the selected samples, the particle filter is used to estimate the different states in the motion model. In equation (1), the term $p(x_t|x_{t-1})$ represents the motion model, which calculates the probability of the state variable at frame *t* based on the state variable at frame *t−1*, showing the state transition between two consecutive frames [2].

3.3.2. Observation Model

The observation model, or appearance model, represents the tracked object and calculates the probability of each of the selected states. In equation (2), the term $p(y_t|x_t)$ represents the observation model [2]. Assuming that the variations in the feature vectors between two consecutive frames are very small, we construct our similarity function based on the correlation method. In other words, in the observation model, the measurement criterion for the selected samples considers the correlation between each of them and the reference pattern. Thus, if the reference pattern is denoted by *t*, and the *j-th* sample of the estimated states and the corresponding observation vector are denoted by x^j and y^j, respectively, the observation model is represented as follows. (For simplicity and without losing generality, the time subscript is omitted).

$$p(y^j|x^j) = corr(t, y^j) \tag{9}$$

Since, in this method, both the reference pattern and the samples are represented as vectors, in the observation model, the correlation is calculated using equation (8), which can be considered as the cosine similarity between the difference vector of the reference pattern from its mean and the difference vector of each sample from its mean. In other words, in equation (8), if the variables *A* and *B* correspond to the reference pattern vector *t* and the selected sample vector *y*, respectively, we have:

$$corr(t, y) = \frac{\sum_k\left((t_k-\mu_t)(y_k-\mu_y)\right)}{\sqrt{(\sum_k(t_k-\mu_t)^2)\left(\sum_k(y_k-\mu_y)^2\right)}} = \frac{\langle(t_k-\mu_t),(y_k-\mu_y)\rangle}{\|(t_k-\mu_t)\|\|(y_k-\mu_y)\|} = s\left((t_k-\mu_t),(y_k-\mu_y)\right) \tag{10}$$

In this equation, ⟨.,.⟩ denotes the dot product of two vectors, and ‖.‖ represents the Euclidean norm of each vector, which is equal to the magnitude of the vector. *s(.,.)* represents the cosine similarity [2] between two vectors, which is actually the cosine of the angle between these two vectors.

3.3.3. Updating the Template

Since in video tracking, changes in the states of the target object may occur, such as approaching or distancing, leading to a change in size, a change in perspective from the front view to a side view, shadow creation or elimination, etc., updating the reference pattern is inevitable. In this method, after obtaining the best selected state in the current frame x_t^* and its corresponding observation model y_t^*, if the correlation between the reference pattern and the selected sample exceeds a certain threshold, the reference pattern is updated as a combination of the current reference pattern and the selected sample, as shown below.

$$\begin{cases} t_{t+1} \leftarrow (1-\eta)t_t + \eta y_t^* & \text{if} \quad corr(t_t, y_t^*) \geq \varepsilon \\ t_{t+1} \leftarrow \mathrm{t}_t & \text{otherwise} \end{cases} \tag{10}$$

4. SIMULATION AND RESULTS

The simulation was conducted using MATLAB 2014a software on a computer with an Intel i5 processor. The number of particles (estimated samples) for the particle filter in the motion model was set to 600 samples. The reference pattern and observation samples were resized to images of 32×32 pixels, and then the described operations were applied to them. In the pattern update step, the threshold ε was set to 0.85, and η was set to 0.95.

4.1. Comparison of Results

To compare the results, metrics such as central error and its average, overlap rate, and its average were used. The following sections explain the mentioned metrics, and then the results will be discussed.

4.1.1. Central Error

Central error is typically defined as the Euclidean distance between the center position of the tracked targets and their correct predefined position. A smaller central error indicates a more accurate tracking result in each frame [2].

4.1.2. Overlap Rate

The overlap rate is used to compare conventional trackers. Given the tracking area in each frame R_T and the correct predefined area R_g, the overlap rate is defined as the ratio of the common area between the tracking and correct areas to their union [17,16,2].

$$Overlap\ Rate = \frac{area(\cap(R_T,R_g))}{area(\cup(R_T,R_g))} \tag{10}$$

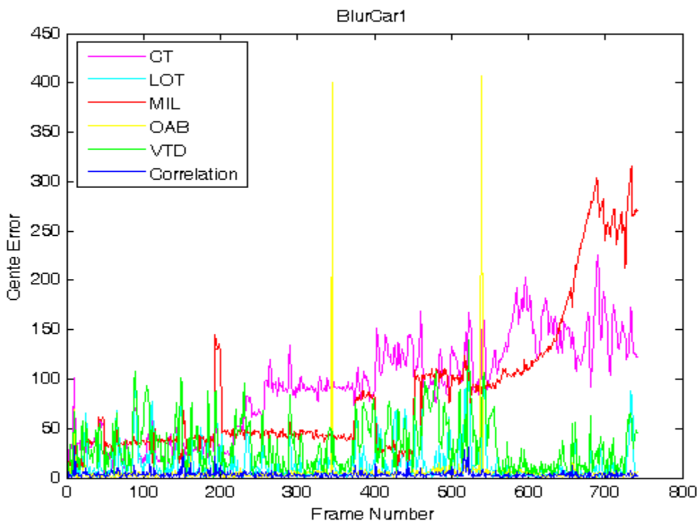
The overlap rate will always be less than or equal to one, and the closer this metric and its average are to one, the better the tracking performance.

4.1.3. Success Rate

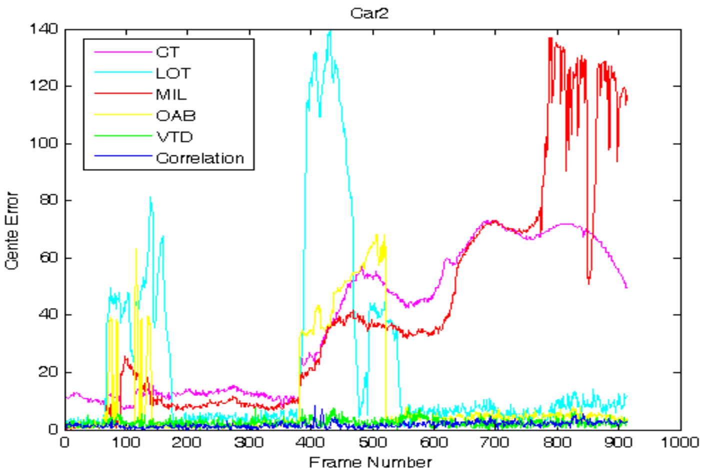
Since the correct area for tracking is manually defined and contains noise, the overlap rate can be overly conservative for the tracking task [2]. Recent studies [17, 16] have used the success rate metric to evaluate the accuracy of tracking algorithms. If the overlap value in each frame is greater than a threshold (e.g., 0.5), the tracking result for that frame is considered a success. A larger average success rate indicates more accurate tracking results.

4.2. Results Analysis

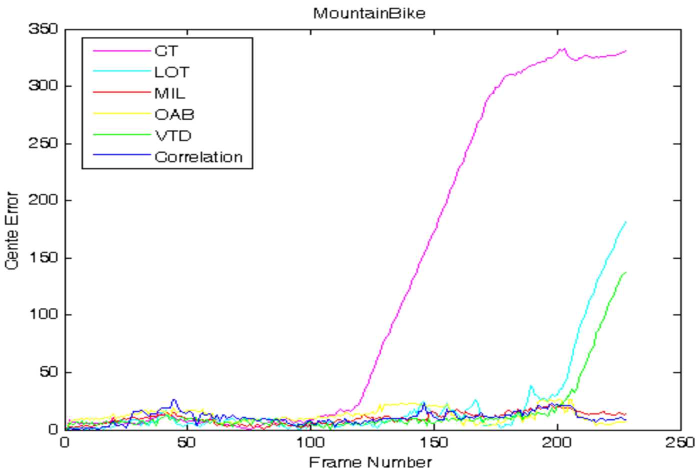
This section reviews and compares the results obtained from the proposed correlation-based method with five other tracking methods, namely CT, LOT, MIL, OAB, and VTD. The evaluation was conducted on four videos: BlurCar1, MountainBike, Car2, and Human7. Figure 2 shows the central error for the four videos mentioned.



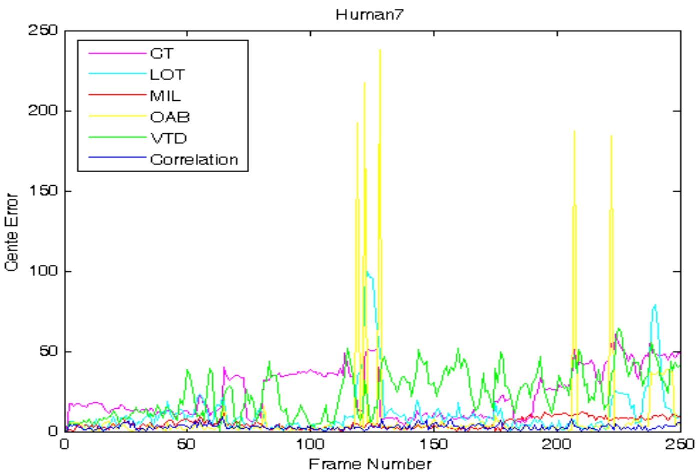
(a)



(b)



(c)



(d)

Fig. 2. Central error graph for the methods CT, LOT, MIL, OAB, VTD, and Correlation (Correlation) on the following databases: a) BlurCar1, b) Car2, c) MountainBike, d) Human7

In this section, we review and compare the results obtained from the proposed correlation-based method with five other tracking methods: CT, LOT, MIL, OAB, and VTD. The evaluation was carried out on four videos: BlurCar1, MountainBike, Car2, and Human7. Figure 2 shows the central error for the four mentioned videos.

In Figure 2-a, the central error graph for tracking methods CT and MIL shows the highest values. VTD and LOT methods follow these two with relatively higher errors. Meanwhile, the OAB method has less error compared to the aforementioned methods, although OAB exhibits the highest error in two specific instances among all methods. The Correlation method has the least central error among the methods tested.

In Figure 2-b, the central error for the tracking methods MIL and CT is the highest. After them, LOT and OAB exhibit more significant errors. Among all the trackers, the Correlation method has the least error, followed by the VTD method with the next lowest error.

In Figure 2-p, the central error values for the mentioned trackers are generally acceptable. However, in the final frames, error values for the LOT and VTD methods significantly increase. The error increase for the CT method starts in the middle frames and peaks in the final frames, with this increase being much higher than that of the LOT and VTD trackers.

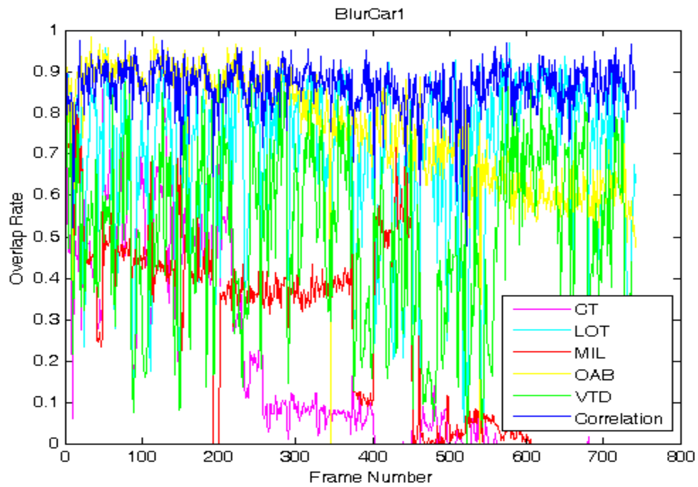
In Figure 2-t, the central error for the VTD tracker exhibits considerable fluctuations, indicating frequent target loss with this method. In the CT method, error increases in specific intervals. The LOT and OAB methods show smaller error values, but in some frames, the error increases. This increase in error is more pronounced and significant in the OAB method compared to LOT, though the number of frames with increased error is higher in LOT than in OAB. The central error for the MIL and Correlation methods is minimal, with the Correlation method exhibiting the least error. Table 1 shows the average central error for the aforementioned trackers across the mentioned videos.

Table 1. Average central error for different trackers

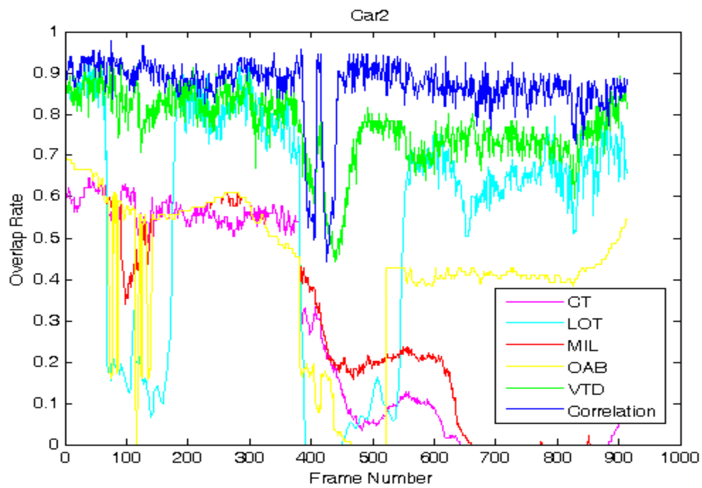
Video	Human7	MountainBike	Car2	BlurCar1
Correlation	2.7522	10.5401	1.7463	3.5838
CT	24.1385	118.9055	37.9458	92.4354
LOT	11.7589	21.9574	22.1336	16.4605
MIL	4.9183	9.9095	40.8430	86.5356
OAB	8.3006	12.6192	11.0236	5.5147
VTD	23.8314	15.9743	2.4741	32.5140

Based on the graphs in Figure 2 and the values in Table 1, it can be concluded that, in terms of central error, tracking using the Correlation method results in the lowest error and therefore exhibits the best performance compared to the other methods mentioned.

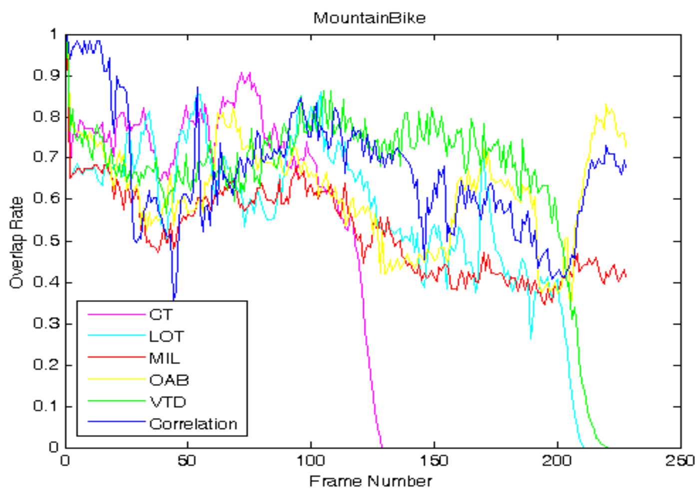
In Figure 3, the overlap rate values for tracking in the given videos using the above methods will be displayed and analyzed.


BlurCar1
Overlap Rate
Frame Number
CT
LOT
MIL
OAB
VTD
Correlation

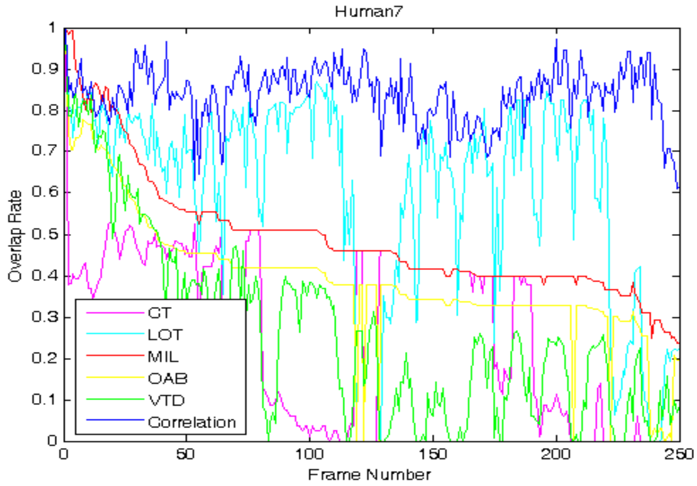
(a)


Car2
Overlap Rate
Frame Number
CT
LOT
MIL
OAB
VTD
Correlation

(b)


MountainBike
Overlap Rate
Frame Number
CT
LOT
MIL
OAB
VTD
Correlation

(c)



(d)

Figure 3: Overlap rate graph for the CT, LOT, MIL, OAB, VTD, and Correlation tracking methods on the following databases: a) BlurCar1, b) Car2, c) MountainBike, d) Human7

Table 2: Average overlap rate for different trackers

Video	Human7	MountainBike	Car2	BlurCar1
Correlation	0.8377	0.6673	0.8670	0.8625
CT	0.2508	0.3941	0.2722	0.1770
LOT	0.6032	0.5394	0.5405	0.6994
MIL	0.4925	0.5214	0.2971	0.2571
OAB	0.3909	0.6198	0.4201	0.7632
VTD	0.2639	0.6433	0.7677	0.5051

The graphs in Figure 3 and the values in Table 2 indicate that in terms of overlap rate and its average, the tracking method using Correlation performs the best compared to the other methods analyzed. This means that, among the listed methods, the tracking area using Correlation is the closest to the target area.

Table 3 shows the average success rate for the above trackers in the given videos. The threshold used to determine tracking success in each frame for all the trackers is set to 0.5.

Table 3. Average success rate for different trackers in percentage

Video	Human7	MountainBike	Car2	BlurCar1
Correlation	100	89.47	99.23	99.86
CT	6.40	51.32	41.51	14.69
LOT	76.00	66.23	69.99	84.37
MIL	41.60	53.07	33.52	11.45
OAB	15.20	82.02	36.69	99.19
VTD	14.80	88.16	98.03	52.96

Table 3 shows that among the methods mentioned, the algorithm used in this paper has also achieved the best performance in terms of average success rate, with values very close to 100. Although some other methods might also show high success rates for certain videos, the Correlation method has achieved the highest values across all the considered videos.

5. CONCLUSION

This paper presented a video tracking method using a particle filter in the motion model and correlation measurement in the observation model. Based on the analysis and the error center graphs, overlap rate, and average values for the error center, overlap rate, and success rate, it can be concluded that the Correlation-based tracking method outperforms the CT, LOT, MIL, OAB, and VTD methods. It achieves the lowest error center and the highest overlap rate and success rate, indicating superior tracking performance compared to the other methods. It should be noted that since the proposed Correlation-based method is computationally efficient, it also offers high tracking speed and can be applied in real-time applications.

Declaration

We acknowledge that we used ChatGPT to enhance the academic writing of our manuscript while ensuring the originality and integrity of our work.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Tang F. and Brennan S. (2007), "Co-Tracking Using Semi-Supervised Support Vector Machines", IEEE 11th International Conference on Computer Vision. DOI: 10.1109/ICCV.2007.4408954

[2] Wang D. and Lu H. (2015), "Visual Tracking via Weighted Local Cosine Similarity", IEEE Trans. on Cybernetics, 45(9), pp. 1838 - 1850. DOI: 10.1109/TCYB.2014.2360924

[3] Wang Q. and Chen F. (2011), "Tracking by Third-Order Tensor Representation", IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(2), pp. 385 - 396. DOI: 10.1109/TSMCB.2010.2056366

[4] Wang D. and Lu H. (2013), "Online Object Tracking With Sparse Prototypes", IEEE Trans. on Image Processing, 22(1), pp. 314 - 325. DOI: 10.1109/TIP.2012.2202677

[5] Zhang K. and Liu Q. (2016), "Robust Visual Tracking via Convolutional Networks Without Training", IEEE Trans. on Image Processing, 25(4), pp. 1779 - 1792. DOI: 10.1109/TIP.2016.2531283

[6] Zhong W. and Lu H. (2014), "Robust Object Tracking via Sparse Collaborative Appearance Model", IEEE Trans. on Image Processing, 23(5), pp. 2356 - 2368. DOI: 10.1109/TIP.2014.2313227

[7] Yang X. and Wang M. (2016), "An Efficient Tracking System by Orthogonalized Templates", IEEE Trans. on Industrial Electronics, 63(5), pp. 3187 - 3197. DOI: 10.1109/TIE.2016.2515559

[8] Zhang K. and Zhang L. (2014), "Fast Compressive Tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, 36(10), pp. 2002 - 2015. DOI: 10.1109/TPAMI.2014.2315808

[9] Widynski N. and Dubuisson S. (2011), "Integration of Fuzzy Spatial Information in Tracking Based on Particle Filtering", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS, 41(3), pp. 635 - 649. DOI: 10.1109/TSMCB.2010.2064767

[10] Yilmaz A. and Javed O. (2006), "Object tracking: A survey", ACM Computing Surveys (CSUR), 38(4). DOI: 10.1145/1177352.1177355

[11] Parekh H.S. and Thakore D.G. (2014), "A Survey on Object Detection and Tracking Methods", International Journal of Innovative Research in Computer and Communication Engineering, 2(2).

[12] Mohanapriya D. and Mahesh K. (2016), "A SURVEY ON VIDEO OBJECT TRACKING SYSTEM", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), 3(20).

[13] Parmar M. (2016), "A Survey of Video Object Tracking Methods", International Journal of Engineering Development and Research (IJEDR), 4(1).

[14] Athanesious MJ. and Suresh P. (2013), "Implementation and Comparison of Kernel and Silhouette Based Object Tracking", International Journal of Advanced Research in Computer Engineering & Technology, p. 1298.

[15] Ross D.A. and Lim J. (2008), "Incremental Learning for Robust Visual Tracking", International Journal of Computer Vision, 77(1-3), pp 125-141. DOI: 10.1007/s11263-007-0075-7

[16] Zhang K. and Zhang L. (2012), "Real-Time Compressive Tracking", European Conference on Computer Vision, pp 864-877. DOI: 10.1007/978-3-642-33712-3_62

[17] He S. and Yang Q. (2013), "Visual Tracking via Locality Sensitive Histograms", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2427-2434. DOI: 10.1109/CVPR.2013.314