Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 157 (2019) 521–529

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12-13 September 2019

Object Tracking Based on Meanshift and Particle-Kalman Filter Algorithm with Multi Features

Irene Anindaputri Iswanto[a], Tan William Choa[b], Bin Li[b,*]

[a]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

[b]*School of Automation Science and Engineering, South China University of Technology, 381 Wushan Rd, Tianhe Qu, Guangdong, Guangzhou, 510630, China*

Abstract

Object tracking is considered to be a key and important task in intelligent video surveillance system. Numerous algorithms were developed for the purpose of tracking, e.g. Kalman Filter, particle-filter, and Meanshift. However, utilizing only one of these algorithms is considered inefficient because all single algorithms have their limitations. We proposed an improved algorithm which combines these three traditional algorithms to cover each algorithms drawbacks. Moreover we also utilized a combination of two features which are color histogram and texture to increase the accuracy. Results show that the method proposed in this paper is robust to cope with numerous issues, e.g. illumination variation, object deformation, non linear movement, similar color interference, and occlusion. Furthermore, our proposed algorithm show better results compare to other comparator algorithms.

Keywords: Multi features; Object Tracking; Meanshift; Particle-Kalman Filter;

1. Introduction

Due to increasing demand of video surveillance system, intelligent video surveillance system has become challenging subject in computer vision. In intelligent video surveillance system there are four key steps, namely object detection, classification, tracking, and analysis [1]. Among these four key steps, object tracking is regarded as a key and crucial task in intelligent video surveillance system.

Over the years, tracking remains an open research discussion in computer vision. Creating a vigorous, efficient, and accurate tracking system remains as an exacting issue up till today. This issue intricacy vastly relies on how to determine the position of the target object over the frames. Difficulties occur in object tracking usually are caused by a number of limitations which either caused by the environment or the characteristic of the target object itself.

* Corresponding author.
E-mail address: binlee@scut.edu.cn

10.1016/j.procs.2019.09.009

Common visual object tracking problem usually comes from the attribute of the moving object itself. In object tracking, the main issue usually come from the changes in the target-objects appearance over time. When object is rigid, the tracking process can take advantages from the accurate knowledge of the objects appearance. However, when the target-object is non-rigid, no accurate appearance model of the target object is available, hence it makes tracking is difficult to perform. Another issue also come from the target objects characteristic, when the target object has fast motion and moves in non-linear motion.

Problem in visual object tracking can also come from the interaction between target-object and other objects, which is called occlusion. Occlusion (either partial or full) may have an impact on the background frames computation process. If the target object has different color distributions with other objects and the background, there will be no problem if only a single visual feature (for example: color) is used to represent a target object. However the possibility of the occurrence of similar color interference is always exist in the video scene. This form of inconsistent transition is always exist as target-objects are commonly moving. Due to this reason, the appearance of the target object may variate over the frames. These issues taken from the three concepts, e.g. variation in object posture or usually called as deformations, illumination, and occlusion[2].

Based on several tracking object challenges above, we are motivated to create a vigorous object tracking system and able to cope with above mentioned problems. In our research, we develop a robust object tracking algorithm by analyzing some conventional tracking algorithms and advantages of each algorithms as a mean to cover each algorithms limitations.

2. Previous Work

Over the years, there are numerous advancement created in object tracking. The algorithms in tracking is roughly classified into 2 types: deterministic and probabilistic method[3].

Due to its easiness and effectiveness, Mean-shift tracking algorithm is popular among numerous deterministic methods. Mean-shift is considered as non-parametric estimation for density which repeatedly finds the highest degree of resemblance between the distribution patterns of candidate region in current frame with target models distribution pattern[4]. Mean-shift is utilized due to its power in dealing with partially occluded object. Moreover, mean-shift also low in complexity that contributes to its efficient computation. Therefore, it is applicable for real time implementation. Mean-shift algorithm also has several limitations despite its numerous advantages. One of the Means-shifts shortcomings is its incapability in tracking object moves in high speed velocity[5]. Furthermore when the target object is occluded severely, Mean-shift often fails to find the object location[6]. The classic mean-shift tracking that utilize color feature[7] shows satisfying result particularly in object with partial occlusion. Nonetheless, when the classic color based means-shift algorithm has to deal with variation in illumination and similar color interference issues, the algorithm often shows failure in tracking the object location accurately. Based on this reason, Hongxia Chu et al.[3] came up with another advance Mean-shift tracking algorithm. Instead of utilizing common histogram to represent the color of each object, Hongxia utilize weighted histogram with spatial corrected background and provide a complete derivation of it. His experimental results show that their proposed algorithm can successfully track the object in the existence of both variation of illumination and occlusion.

Another methods in tracking is probabilistic method, e.g. Kalman Filter and Particle Filter. Kalman Filter is used to provide optimal estimation for linear system by providing solution recursively[8]. The position of a constant moving occluded object also can be predicted using the prediction system in Kalman Filter. However, this can bring a problem when the object has non-linear motion. Ravi Pratap Tripathi et al. utilize the combination of 3 algorithms, e.g. memory attenuation, optimal likelihood prediction, and robust innovation prediction. The implementation of combined algorithms shows a remarkable result in real time object tracking.[9]. Contrast to the Kalman Filter and Extended Kalman Filter tracking algorithm, the combined algorithm created by Ravi shows better performance in both accuracy and reliability. tracking algorithm. Another tracking algorithm is proposed by Nima et al. which utilize new observation model method for Kalman Filter algorithm[10]. For the observation model, the algorithm utilize Gaussian function to give weight variation for background and foreground regions. As the result, the performance of Nima model can surpass the accuracy of other methods.

Another example of probabilistic method is Particle Filter. In contrast to Kalman Filter algorithm, Particle Filter is effective for solving state space prediction. To define the probability distribution, particle filter employ unsystematic samples called as "particle", where each particle is associated with a certain weight. Then the estimation will be computed based on these samples and weights[11]. Besides its capability to deal with non-Gaussian problem, Particle Filter is also able to deal with severe occlusion and recover from losing track which is usually happened because of full occlusion[12]. However how to reduce the computation cost (reducing the number of particles) is still a challenging task in tracking with particle[13]. Xinting et al. utilized a set of indications as well as another four models, e.g. Gaussian appearance spatial color mixture, elliptical shape combined with a likelihood function, a motion estimation model, and a model of Edge Orientation Histogram. This technique shows a good performance result of Particle Filter to track occluded object[14]. However, Particle Filter also have some disadvantages. The main disadvantages of this algorithm exist in the need of significant number of samples or particles. Each particle samples in Particle Filter may have low contribution and efficiency, hence to overcome this problem a significant amount of particles are needed to perform an accurate estimation. The big number of particles will increase the complexity and contribute to the low system performance. Thus it makes Particle Filter not applicable for real time implementation.

Based on the recent works in object tracking that have been mentioned previously, it can be concluded that to develop more effective and efficient tracking system, utilizing only one single algorithm is not enough. Based on these reasons, some researchers combined different visual object tracking algorithm to deal with various conditions while still maintaining its accuracy. Irene[15] came up with a fusion of Mean-shift and Particle-Kalman Filter, called as Mean-shift Particle-Kalman Filter (MPKF) tracking algorithm. In her method, color based Mean-shift is utilized when the object moves linearly without occlusion. In contrast, when the object is occluded or the result of mean-shift tracker is unconvincing, color-based Particle-Kalman Filter (PKF) is employed to enhance the performance. The experiment outcome show that the system can track target-object under occlusion and increase the systems speed performance compare to other tracking algorithms combination (i.e.Mean-shift Particle Filter, Mean-shift Kalman filter)[15]. However, because Irene only utilize a single feature e.g. color histogram, the system is unable to track the object when similar color interference occurred.

In this paper a new multi-features-based object tracking algorithm is proposed. In our proposed method, color information and texture information are used to define the target-object. The combination of HSV color based Mean-shift and color-texture based Particle-Kalman Filter (PKF) are used interchangeably as the main tracker for the system. To resolve the issues concerning on the appearance of the target-objects caused by variation in illuminations, the target model is updated in every frame when target-object is visible in the scene or when Mean-shift algorithm is used in the system.

3. The Proposed Method

In this section, a short review of the proposed algorithm with flowchart is given. The flowchart can be seen in Fig.1.

3.1. Target Model Initialization

Because our work will be more focus on developing a robust tracking algorithm, in our system, the target object is determined manually by selecting the RoI - Region of Interest (represented by a rectangular area) from the first frame of the video input.

From the targeted object, two features are extracted, color and texture. This part will be explained in detail in the following subsection.

3.1.1. Color Feature Extraction (HSV Weighted Color Histogram

Based on our previous work [], for color feature extraction we still use 8x8x4 HSV weighted histogram proposed by Commaniciu et al[7] to reduce the effects of background and make the distribution more reliable and effective.

$$q(u) = C \sum_{i=1}^{n} \left(\left\| \frac{x_c - x_i}{h} \right\| \right)^2 \delta(b(x_i) - u) \tag{1}$$

Where x_c defines the midpoint of the region of interest, C is the normalized coefficient, and h is the normalizing constant ($h = \sqrt{H_x^2 + H_y^2}$).

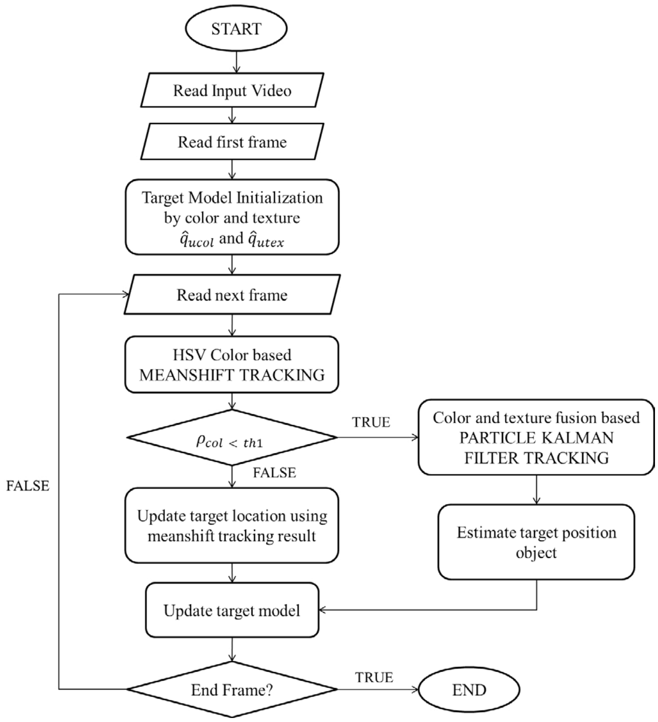

Fig. 1: Flowchart of the proposed algorithm

3.1.2. Texture Feature Extraction (LBP Texture Histogram

The Local Binary Pattern (LBP) operator is an image operator that represents the picture texture into an array. After allocation of image as an array, the histogram is used for further analysis. The working principle of LBP operator is to first set up the threshold value of neighboring pixel and transforming them into a binary value, after obtaining the binary value it is further converted to decimal number. LBP operator is expressed below:

$$LBP_{P,R}(X_C, Y_C) = \sum_{P=0}^{P-1} S(g_p - g_c)2^P \tag{2}$$

Where $S(x) =$, g_c is neighborhood center pixel (x_c, y_c) where P refers to Pixel and R refers to the Radius. g_p is the grey value of the neighboring pixels around the center pixel. the value of P R is chosen by selves P=8 and R=1.

Timo Ojala et al. describe the redefined LBP pattern as $LBP^{riu2}_{P,R}$ which combines the rotational and uniform patterns [16]. The symbol $riu2$ refers to the rotation invariant of the LBP patterns and the conversion value between 1 and 0 are less than twice spatially. Different labels are set for each uniform pattern and the entire non-uniform patterns are assigned to label (P+1). The following formula defines the rotation invariant uniform LBP patterns.

$$LBP^{riu2}_{P,R} = \begin{cases} \sum_{P=0}^{P} s(g_p - g_c), & (\mathrm{LBP}_{P,R}) < 2P + 1, \\ other \end{cases} \tag{3}$$

$$U(LBP_{P,R} = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{p-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{4}$$

In $LBP^{riu2}_{8,R}$ operator, there are 8 different patterns altogether, each patterns are assigned to the different task for instance pattern 0 and 1 is responsible for noise, 7 and 8 represent dim spots, 2 and 6 define the line , 3 and 5 define the edges, and pattern 4 defines the outer line or boundary. For the convenient, these patterns are collectively divided in accordance to their task (carrying the information). For instance pattern 0, 1, 7, and 8 are grouped as minor patterns remaining other five patterns (pattern 2, 3, 4, 5, and 6) are defined as major patterns. Furthermore these five patterns which are assigned as the major groups are entitled for executing the texture information for the proposed algorithm.

3.1.3. Occlusion Detection

Occlusion detection is done by calculating the similarity between objects using similarity function from our previous work. If the similarity function does not fulfill the requirement of occlusion ($\rho_c ol < th_1$) it means that there is no occlusion occurred. When there is no existence of occlusion, the target location will be retrieved by Mean-shift tracker based on HSV color feature. In contrast, when occlusion is found in the scene, the tracker will be swapped into Particle-Kalman Filter.

3.1.4. Update Target Model

We also still use the same update model equation as our previous work, as follows:

$$q_k^{(u)} = (1-\alpha)q_{k-1}^u + \alpha p_{E(X_k)}^{(u)} \tag{5}$$

where in our experiment we determined the value of α to be 0.15.

3.1.5. Color and Texture Features Fusion Based Particle-Kalman Filter Tracking

As explained before, our proposed algorithm, color and texture feature fusion based Particle Filter which includes the principle of Kalman Filter is applied when the Mean-shift tracking result shows unconvincing result or when occlusion occurred. Similar with conventional Particle Filter, the distribution of target-object is estimated by a group of particles and represented by

$$S_k^{(i)} = \left\{X_k^i, w_k^{(i)}\right\}_{(i=1..N)} \tag{6}$$

where,

$X_k^{(i)}$: hypothetical state of each particle's tracked object position at frame k

$w_k^{(i)}$: hypothetical state of each particle's tracked object weights at frame k

To ensure the accuracy of the measurement process, a small noise is assigned for half of particles and bigger noise for remaining particles. After the occurrence of full occlusion, the particle assigned with big noise is used for tracking the object, as shown in the equation 7 and 8

$$\hat{X}_k^{(i:(1/2*N))} = \hat{X}_{k-1}^{(i:(1/2*N))} + \omega_{k(small}^{(i:(1/2*N))} \tag{7}$$

$$\hat{X}_k^{(i:(1/2*N))} = \hat{X}_{k-1}^{(i:(1/2*N))} + \omega_{k(big}^{(i:(1/2*N))} \tag{8}$$

Meanwhile, each particle weight is calculated by the weighted fusion formula as follows

$$\hat{W}_k^{(i)} = \alpha_p \hat{W}_{kcol}^{(i)} + \beta_p \hat{W}_{ktex}^{(i)} \tag{9}$$

$$\alpha_p = \frac{\sum_{i=1}^N \hat{W}_{kcol}^{(i)}}{\sum_{i=1}^N \hat{W}_{kcol}^{(i)} + \sum_{i=1}^N \hat{W}_{ktex}^{(i)}} \tag{10}$$

$$\beta_p = \frac{\sum_{i=1}^N \hat{W}_{ktex}^{(i)}}{\sum_{i=1}^N \hat{W}_{kcol}^{(i)} + \sum_{i=1}^N \hat{W}_{ktex}^{(i)}} \tag{11}$$

$$\hat{W}_{kcol}^{(i)} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1-\rho_{col}\left\{\hat{p}_{ucol}(\hat{X}_k^{(i)}), \hat{q}_{ucol}\right\}}{2\delta^2}\right) \tag{12}$$

$$\hat{W}_{ktex}^{(i)} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1-\rho_{tex}\left\{\hat{p}_{utex}(\hat{X}_k^{(i)}), \hat{q}_{utex}\right\}}{2\delta^2}\right) \tag{13}$$

$$\rho_{col}\left[\hat{p}_{ucol}\left(\hat{X}_k^{(i)}\right), \hat{q}_{ucol}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_{ucol}(\hat{X}_k^{(i)}) * \hat{q}_{ucol}} \tag{14}$$

$$\rho_{tex}\left[\hat{p}_{utex}\left(\hat{X}_k^{(i)}\right), \hat{q}_{utex}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_{utex}(\hat{X}_k^{(i)}) * \hat{q}_{utex}} \tag{15}$$

where,
$\hat{W}_{k(i)}$: comprehensive weight of the i-th predicted particle
$\hat{W}_{kcol}^{(i)}$: weight of the i-th predicted particle based on color feature
$\hat{W}_{ktex}^{(i)}$: weight of the i-th predicted particle based on texture feature
αp : the fusion coefficient of color feature in prediction phase
βp : the fusion coefficient of texture feature in prediction phase
$\hat{p}_{ucol}$ and $\hat{q}_{ucol}$: color distribution of the candidate and target model respectively
$\hat{p}_{utex}$ and $\hat{q}_{utex}$: texture distribution of the candidate target model respectively

4. Experiment Results

The algorithm proposed in this paper and other algorithms as comparison are evaluated by implementation on the MATLAB R2014b platform. All experiments were tested on an Intel Core i7-2630QM CPU @2.0GHz, 8GB of RAM and 2GB NVIDIA GEFORCE GT540M Graphics.

4.1. Qualitative Evaluation

The algorithms are implemented on six video datasets[17 18]. The algorithms used for comparison in our experiments are Mean-shift Kalman Filter, Mean-shift Particle Filter, and Mean-shift Particle-Kalman Filter. We present the result of qualitative evaluation in 2 main issues: similar color interference and occlusion.

4.1.1. Similar Color Interference

In this section, the robustness of the algorithm proposed in this paper against similar color interference is presented. To prove the result, two videos from[18] are used, i.e. Subway and Jogging.

Figure 2 shows the tracking results of each tracking algorithms under similar color interference. As we can see, other algorithms fail to track with similar color properties. On the contrary, the tracking results show the proposed method is robust to track target object under similar color interference.

4.1.2. Occlusion

Occlusion is the toughest challenge in object tracking. When only some parts of the target-object are occluded by other object, it is called partial occlusion. Whereas in complete/full occlusion, the target-object is completely invisible knowing that the target-object is still present at the scene.

From the figure 3, we can see that the person in white shirt is the target selected to track. At the beginning, all algorithms are able to detect the the target position. However, after the target is occluded by the pole which has similar color with the target object, the proposed algorithm is the only one that is able to show the accurate tracking result.

4.2. Quantitative Analysis

For the purpose of performing quantitative analysis to the proposed system and other comparison algorithms, a certain set of evaluation metrics is utilized. The results of quantitative analysis using evaluation metrics[19] is shown in the Table 1. As shown in Table 1 we can see that among several tracking algorithms tested in this experiment, Mean-shift Kalman Filter algorithm has the fastest system speed performance due to its low complexity algorithm. On the other hand, our proposed algorithm has the slowest system speed performance compare to other algorithms. From this we can conclude that using multi features to represent the target-object has resulted in the growth of the complexity and the computational load of the proposed algorithm, therefore it will lower the system speed performance of the proposed method.However our proposed algorithm has the highest performance evaluation results in tracker detection rate (TRDR) and accuracy (Acc) in all eight video datasets.In conclusion, our proposed algorithm has higher accuracy and more accurate tracking estimation compare to other algorithms.

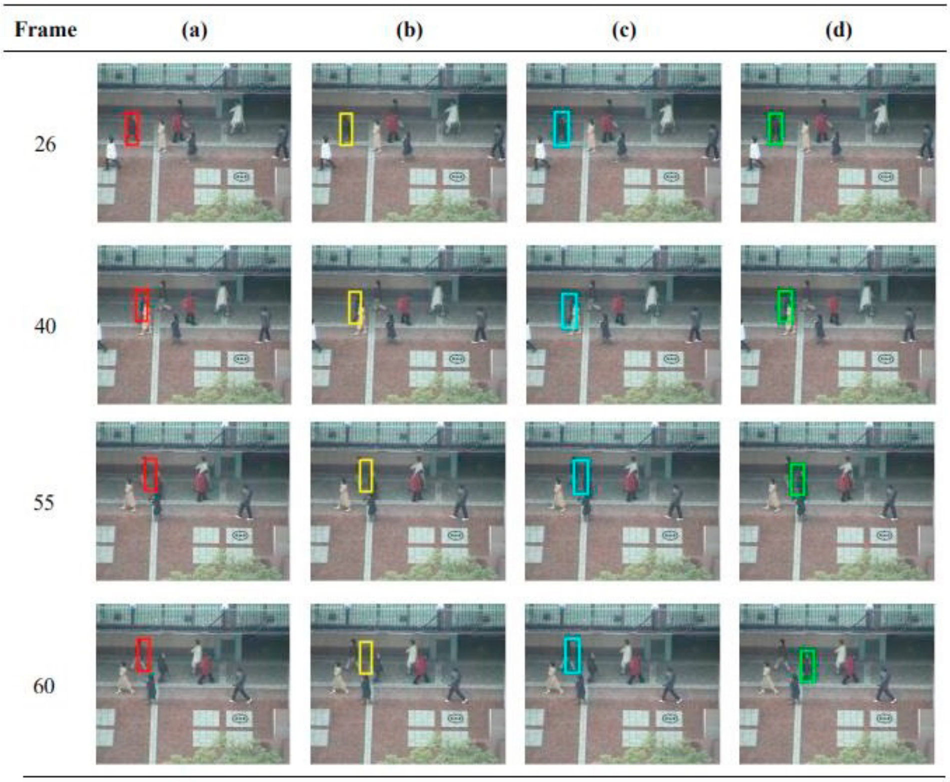

Fig. 2: Tracking results for object under similar color interference for Subway video sequence: (a) Mean-shift Kalman Filter, (b) Mean-shift Particle Filter, (c) HSV color based Mean-shift Particle-Kalman Filter [8], (d) Proposed algorithm

5. Conclusion

In this paper, a robust tracking system that utilize the combination of 3 classic algorithms e.g. Mean-shift, Kalman Filter, and Particle Filter with a fusion of color and texture features is established. The three classic algorithms are combined to elicit the potential of each algorithm and conceal each limitation.

The experiment results prove that the tracking system proposed in this paper is able to be implemented in single object tracking and is able to deal with a number of issues such as illumination variation, object deformation, object with non linear movement, similar color interference, and occlusion. Furthermore, the experiment results show that the proposed tracking system which utilizes multi features to represent the target model has better performance evaluation results in contrast to other comparison algorithms.

References

1. Joshi, K.A., Thakore, D.G.. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering* 2012;**2**(3):44–48.
2. Shaikh, S.H., Saeed, K., Chaki, N.. Moving object detection approaches, challenges and object tracking. In: *Moving object detection using background subtraction*. Springer; 2014, p. 5–14.
3. Chu, H., Song, Q., Yuan, H., Xie, Z., Zhang, R., Jiang, W.. Research of mean shift target tracking with spatiogram corrected background-weighted histogram. In: *2015 IEEE International Conference on Information and Automation*. IEEE; 2015, p. 1942–1946.
4. ComaniciuD, M.. Arobust approach toward feature space analysis [j]. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2002; **24**(5):313–329.
5. Medouakh, S., Boumehraz, M., Terki, N.. Improved object tracking via joint color-lpq texture histogram based mean shift algorithm. *Signal, Image and Video Processing* 2018;**12**(3):583–590.

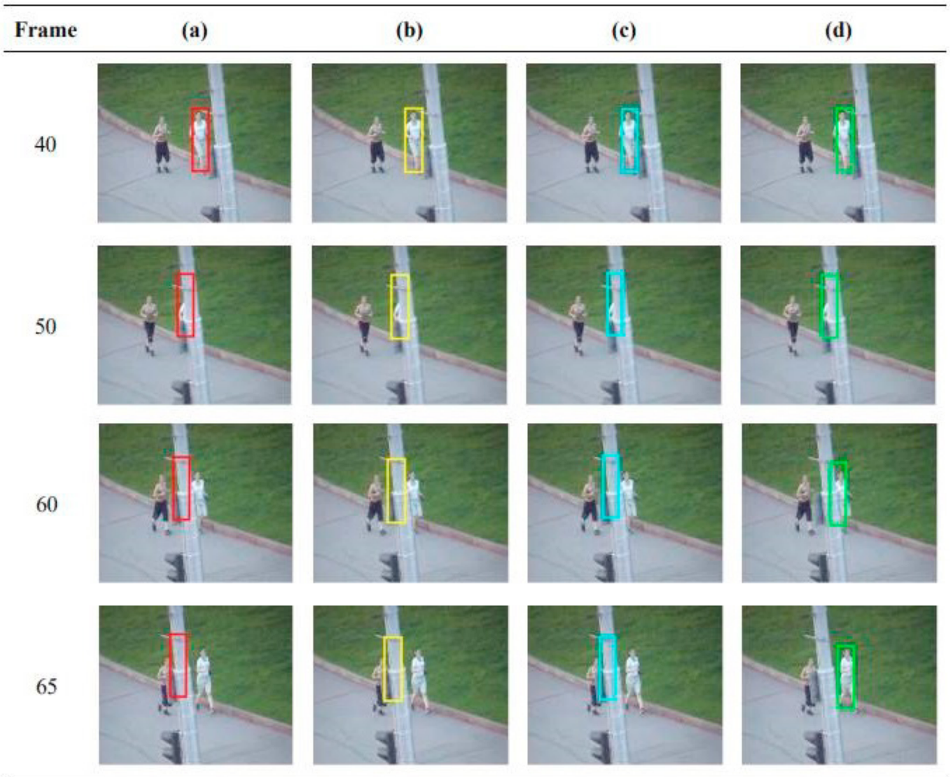

Fig. 3: Tracking results for object under similar color interference and full occlusion for Jogging video sequence: (a) Mean-shift Kalman Filter, (b) Mean-shift Particle Filter, (c) HSV color based Mean-shift Particle-Kalman Filter [8], (d) Proposed algorithm

6. Dulai, A., Stathaki, T.. Mean shift tracking through scale and occlusion. *IET signal processing* 2012;**6**(5):534–540.
7. Comaniciu, D., Ramesh, V., Meer, P.. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 2003;(5):564–575.
8. Welch, G.F.. Kalman filter. *Computer Vision: A Reference Guide* 2014;:435–437.
9. Tripathi, R.P., Ghosh, S., Chandle, J.. Tracking of object using optimal adaptive kalman filter. In: *2016 IEEE International Conference on Engineering and Technology (ICETECH)*. IEEE; 2016, p. 1128–1131.
10. Najafzadeh, N., Fotouhi, M., Kasaei, S.. Object tracking using kalman filter with adaptive sampled histogram. In: *2015 23rd Iranian Conference on Electrical Engineering*. IEEE; 2015, p. 781–786.
11. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing* 2002;**50**(2):174–188.
12. Sigalas, M., Pateraki, M., Trahanias, P.. Full-body pose trackingthe top view reprojection approach. *IEEE transactions on pattern analysis and machine intelligence* 2015;**38**(8):1569–1582.
13. Fang, Y., Wang, C., Yao, W., Zhao, X., Zhao, H., Zha, H.. On-road vehicle tracking using part-based particle filter. *IEEE Transactions on Intelligent Transportation Systems* 2019;.
14. Pan, X., Chen, X., Men, A.. Occlusion handling based on particle filter in surveillance system. In: *2010 Second International Conference on Computer Modeling and Simulation*; vol. 1. IEEE; 2010, p. 179–183.
15. Iswanto, I.A., Li, B.. Visual object tracking based on mean-shift and particle-kalman filter. *Procedia computer science* 2017;**116**:587–595.
16. Ojala, T., Pietikäinen, M., Mäenpää, T.. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 2002;(7):971–987.
17. Kalal, Z.. Tld dataset. 2010.
18. Hanyang, U.. Tracker_benchmark_v1.0. 2013. URL http://cvlab.hanyang.ac.kr/tracker_benchmark/index.html.
19. Bashir, F., Porikli, F.. Performance evaluation of object detection and tracking systems. In: *Proceedings 9th IEEE International Workshop on PETS*. 2006, p. 7–14.

Table 1: Performance Evaluation Results of Several Tracking Algorithms

No.	Dataset	Metrics	Tracking Method			
			MKF	MPF	MPKF	Proposed Algorithm
1	Basketball	TRDR	1	0.9901 ± 0.0029	1	1
		ACC	1	0.9901 ± 0.0029	1	1
		Time(s)	0.0863 ± 0.0002	0.1350 ± 0.0010	0.2254 ± 0.0062	0.1926 ± 0.0046
2	David3	TRDR	1	1	1	1
		ACC	1	1	1	1
		Time(s)	0.1011 ± 0.0041	0.1669 ± 0.0035	0.1248 ± 0.0015	0.2142 ± 0.0013
3	Jogging1	TRDR	0.9771 ± 0.0043	0.9484 ± 0.0220	0.9151 ± 0.0110	0.9876 ± 0.0026
		ACC	0.9788 ± 0.0060	0.9501 ± 0.0237	0.9168 ± 0.0127	0.9893 ± 0.0043
		Time(s)	0.0791 ± 0.0009	0.0840 ± 0.0099	0.2075 ± 0.0010	0.1598 ± 0.0012
4	Jogging2	TRDR	0.1830 ± 0.0095	0.7981 ± 0.0038	0.5528 ± 0.0658	0.9980 ± 0.0020
		ACC	0.1847 ± 0.0112	0.7998 ± 0.0055	0.5545 ± 0.0675	0.9988 ± 0.0037
		Time(s)	0.0744 ± 0.0006	0.0855 ± 0.0008	0.0880 ± 0.0042	0.1576 ± 0.0048
5	Woman	TRDR	0.7255 ± 0.0174	0.9309 ± 0.0087	0.9310 ± 0.0011	0.9957 ± 0.0029
		ACC	0.7255 ± 0.0174	0.9309 ± 0.0087	0.9310 ± 0.0011	0.9957 ± 0.0029
		Time(s)	0.0856 ± 0.0015	0.1446 ± 0.0013	0.1198 ± 0.0021	0.1935 ± 0.0042
6	Subway	TRDR	0.3092 ± 0.0066	0.2276 ± 0.0029	0.8138 ± 0.0072	0.9318 ± 0.0086
		ACC	0.3092 ± 0.0066	0.2276 ± 0.0029	0.8138 ± 0.0072	0.9318 ± 0.0086
		Time(s)	0.0754 ± 0.0002	0.0739 ± 0.0002	0.0913 ± 0.0006	0.1010 ± 0.0006
7	CarChase1	TRDR	0.5247 ± 0.0075	0.6027 ± 0.0357	0.8225 ± 0.0170	0.9901 ± 0.0067
		ACC	0.5661 ± 0.0066	0.6376 ± 0.0326	0.8749 ± 0.0155	0.9910 ± 0.0192
		Time(s)	0.1164 ± 0.0023	0.1027 ± 0.0061	0.1500 ± 0.0056	0.1731 ± 0.0194
8	CarChase2	TRDR	0.8061 ± 0.0832	0.7067 ± 0.0060	0.9049 ± 0.0011	0.9866 ± 0.0043
		ACC	0.8185 ± 0.0777	0.7273 ± 0.0056	0.8859 ± 0.0078	0.9876 ± 0.0040
		Time(s)	0.1131 ± 0.009	0.1201 ± 0.0051	0.1422 ± 0.0315	0.1815 ± 0.0038