Journal of Computer & Robotics 12 (1), 2019 93-101

Computer & Robotics

A New Approach to Improve Tracking Performance of Moving Objects with Partial Occlusion

Zahra Sahraei, Amir Masoud Eftekhari Moghadam *

Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Received 20 June 2019; Revised 05 August 2019; Accepted 24 October 2019; Available online 07 January 2020

Abstract

Tracking objects in video images has attracted much attention by machine vision and image processing researchers in recent years. Due to the importance of the subject, this paper presents a method for improving object tracking tasks with partial occlusion, which increases the efficiency of tracking. The proposed approach first performs a pre-processing and extracts the tracking targets from the image. Then the salient feature points are extracted from the targets that are moving objects. In the next step, the particle filter is used for tracking. The final steps are modifying points and updates. A new approach is used to determine the speed of the feature points because the speed of some points can be out of range and this causes errors in tracking especially when there is occlusion. The location of the new points is corrected and updated using the threshold values in modifying the process as needed. The experiments performed on the video sequence of PETS2000 database show that the precision and recall of the proposed approach are higher than other compared approaches.

Keywords: *Particle Filter, Salient Feature Points, Partial Occlusion, Object Tracking.*

1. Introduction

Moving object tracking is a complicated task in computer vision in recent years. The goal of object tracking is an estimation of object location and motion parameters regarding the initial location of the objects [1]. The availability of high-quality video cameras, Emergence of high-speed computers, and the vast need for automatic video analysis have made much interest in visual tracking algorithms. Visual tracking is used in motion detection, auto-monitoring, human-computer interaction, vehicle routing, video indexing, and so on. Noise in images, crowded backgrounds, objects complex movements, partial or complete occlusion, illumination changes, real-time processing and etc. are among the most challenging issues in tracking objects.

Several methods and algorithms are used to handle the occlusion problem and the high speed of objects, each one being able to manage and resolve some of the problems caused by occlusion and speed, however there isn't a single method which has been able to completely resolve the problem of occlusion and speed in all video sequences with different scenarios and to perform tracking without problems.

A method for tracking multiple objects with partial occlusion is provided in this paper. The overall structure of the suggested approach includes four stages. In the first stage, the pre-processing phase and then the feature extraction is done. The third stage is the tracking process. In the final stage modifying points and updates are performed.

* Corresponding author. Email: eftekhari@qiau.ac.ir

2. Literature Review

Typically, two methods are used to track multiple objects. Low-level techniques for detecting size, location and speed of the object, and high-level techniques related to the appearance of the object. In this approach [2], in a video sequence tracking an object is provided by a moving camera. This approach is based on the particle filter and is persistent to other objects with analogous look to the target object. The proposed method clearly recognizes objects that are similar to the target and around it. Tracking and positioning are used to calculate the probability after detecting objects. This method cannot detect all areas of an object similar to a target object in one of the scenes, but it can partly obtain it because it is much bigger than the rectangle size associated with the target object and the similar object. A method is proposed for tracking multiple objects based on a hierarchical framework. This method includes object detection, low-level tracking based on adaptive filters, high-level tracking using several histograms and event management that can detect any collision event by objects and improve tracking under occlusion issue for any appeared object in the scene [3]. In [4] a particle filter is used to track a target object using a rectangular box. The tracking efficiency is increased by an incremental probability function, which is a mixture of similarity calculation and a histogram. This method provides a very fast performance versus precision reduction.

3. Proposed Method

In general, the suggested approach consists of four stages. The first step is pre-processing which involves extracting the background and drawing a bounding box around the objects. The second step is feature extraction, which uses a corner detection algorithm to extract the corners of objects in order to obtain salient feature points (SFPs) for obtaining features from them. In the third step the particle filter is used to track the objects, and then the particles with the highest weight are selected to replace the SFPs. The final step is modifying and updating points. The main idea is that the location of new points of a particle with the maximal weight (points with occlusion, outlier ones and some points with irrational speed) are corrected and updated if needed. Then after updating the bounding box the outliers are corrected. Points with occlusion are also diagnosed and modified. Then the tracked SFPs are updated. Flowchart of the suggested approach has been shown in Fig. 1.

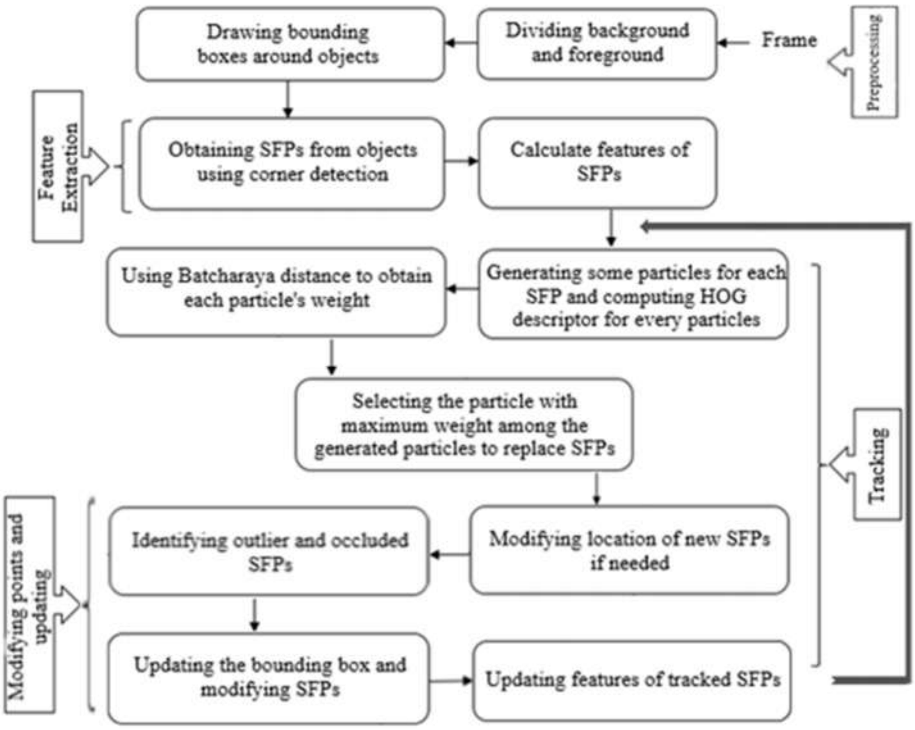

Fig. 1. flowchart of the proposed method

3.1. Pre-processing

This stage involves extracting the foreground and drawing a bounding box around objects. Feature extraction will be performed when the foreground is extracted and the bounding boxes are drown [5].

3.2. Feature Extraction

This stage involves detecting corners as salient feature points, and extracting the features of these points. Harris corner detection algorithm is used to extract corners in this stage [6]. An object is displayed using a set of SFPs. These points represent key points in the object and play a significant role in displaying and tracking moving objects [5].

3.2.1.Feature Calculation of Objects

It is usually determined by a rectangular box, the size and location of each target. An object (O) is shown by a collection of features including the bounding box(BB), velocity$(\mathrm{v_o})$, and a collection of SFPs. Since a collection of SFPs is used to show an object, each SFP contains several attributes: relative location(rl), descriptor(hog), location $(\mathrm{p_l})$, outlier and overlapped flags(f) and velocity of feature points$(\mathrm{v_{fp}})$ [5].

In this paper, four mathematical vector operators are defined, including (addition $\oplus$, subtraction !, multiplication $\otimes$, and division %), and two logical operators ($\leq$ and $\geq$) for the mathematical expressions used. The bounding box of an object consists of size $\mathrm{s} = (\mathrm{width}, \mathrm{height})$ and coordinates $\mathrm{lt} = (\mathrm{x}, \mathrm{y})$ (left-top corner) [5]. The velocity$(\mathrm{v_o})$ of a target object (O) is defined by moving the left-top corner of the bounding box:

$$v_o = BB^{\mathrm{t}}.\,lt\ !\ \ BB^{\mathrm{t\text{-}1}}.\,lt \tag{1}$$

BB^{t} and $BB^{\mathrm{t-1}}$ are bounding boxes of the object O in frames t and $(\mathrm{t}\text{ - }1)$ respectively. $BB^{\mathrm{t}}.\,lt$ and $BB^{\mathrm{t\text{-}1}}.\,lt$ are coordinates of left-top corner of the bounding boxes in frames t and $(\mathrm{t}\text{ - }1)$. The status of an $\mathrm{SFP}(\mathrm{SFP_j})$ is shown by $\{\mathrm{hog}, \mathrm{v_{fp}}, \mathrm{rl}, \mathrm{p_l}\}$. p_l is the location vector of the SFP in the current frame. Relative location rl is defined to show the location vector of SFP regarding to the left-top corner of the bounding box .The left-top corner of the bounding box in the current frame is $B\ .\ lt$ [5].

$$SFP_j\,.\,rl = SFP_j\ .\,p_l\ !\ \ BB\ .\,lt \tag{2}$$

Descriptor (hog) describes the properties of a SFP. In this method, the Histogram of Oriented Gradient (HOG) [7] is applied which is broadly used in many computer vision algorithms [5, 8, 9].

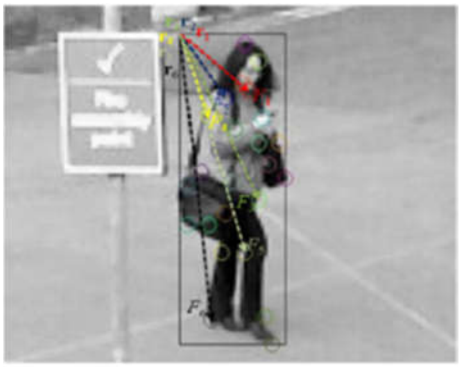

Fig. 2. Relative location of the feature points: The center of each circle indicates the location of the corresponding feature. Vectors depicted in the image represent relative location (rl) of a feature point $(\mathrm{SFP_j})$ regarding the left-top corner of the bounding box [5].

3.2.2. Velocity Calculation of SFP

Calculating the velocity of an overlapping SFP may not be exact, because the SFP may not be evident [5]. Hence, if the SFP is not overlapped, its location difference in two consecutive frames is considered as the velocity; otherwise the movement of the left-top corner of the bounding box is taken as the velocity of an overlapped SFP [5]:

$$SFP_j\,.\,v_{fp} = \begin{cases} SFP_j\,.\,p_l^{\ \mathrm{t}}\ !\ \ SFP_j\,.\,p_l^{\ \mathrm{t\text{-}1}}, & if\ \ SFP_j\,.\,Overlapped = 0 \\ BB^t.\,lt\ !\ \ BB^{t-1}.\,lt, & otherwise \end{cases} \tag{3}$$

$SFP_j\,.\,p_l^{\ \mathrm{t}}$ and $SFP_j\,.\,p_l^{\ \mathrm{t\text{-}1}}$ are locations of the SFP in frames t and $(\mathrm{t}\text{ - }1)$ respectively.

3.3. Tracking

Tracking based on the particle filter estimates the alternative distribution of the target object's location in a frame according to the data from past observations. In this paper, each SFP of an object is tracked and tracking outputs are used to predict the object location in the next frame. Then the SFP features are updated based on their current location [5] .

3.3.1. Particles

A set $\mathrm{PRT} = \{ \mathrm{prt}_k | \mathrm{k} = 1,2,3,\ldots,\mathrm{n}\}$ including n particles for each SFP is created based on its current state to track objects in a frame. Probable places of SFPs in video frames are predicted by generated particles [5]. A particle $\mathrm{PRT_i}$ is produced for each $\mathrm{SFP_j}$ and object O using equation (4):

$$SFP_{\mathrm{j}} \ . prt_k \ . q = SFP_{\mathrm{j}} \ . p_l \oplus SFP_{\mathrm{j}} \ . v_{fp} \oplus N\left(0, d^2\right) \quad (4)$$

$SFP_j \ . v_{fp}$ is velocity which is obtained using equation (3). $N\left(0 \ , d^2\right)$ is Gaussian distribution with zero mean and variance d^2. Then the features of particles are extracted using histogram of oriented gradients (HOG) $\left(SFP_j . prt_k . hog\right)$ in each estimated location $\left(SFP_j . prt_k . q\right)$ [5]. After that, the Bhattacharyya distance is used to obtain the particle's weight [10]:

$$SFP_j \ . prt_k \ . w = BD\left(SFP_j \ . hog \ , SFP_j \ . prt_k \ . hog\right) \quad (5)$$

The Bhattacharyya distance is indicated by $BD(x,y)$ which shows the similarity between two distributions X and Y. $\left(SFP_j . hog\right)$ is the feature descriptor (HOG) for tracked SFP and $\left(SFP_j . prt_k . hog\right)$ is the feature descriptor for estimated particle.

It is based on the resemblance between the tracked SFP and the particle to assign the weight of a particle, therefore a particle is more likely to show the same SFP to the tracked SFP in the current frame when a higher weight is appointed to a particle. One of the old approaches in particle filter is to consider the mean weighted predictions that is performed by n particles, but the location obtained by this method may not always be the exact location of an SFP. So each tracker selects the particle with the maximal weight among the particles produced, because it is necessary to predict the precise location of an SFP in this algorithm. The predicted location for the SFP tracked by this particle is shown [5]:

$$SFP_j \ . p_l = prt_x, \ where \ x = \underset{i \in N}{\arg\max} \ w_i \quad (6)$$

$SFP_j \ . p_l$ is the predicted location of the selected particle with the most weight. The predicted locations of an SFP using a particle with maximal weight and mean weight of particles are shown in Fig 3. The maximal particle weight indicates by a red dot, while the blue point indicates the location of the mean weight of all particles. The red particle represents the SFP with greater accuracy than the mean weight of all particles.

(a) (b)

Fig. 3. Predicted locations of an SFP using a particle with maximal weight and mean weight particles: (a) the main location of an SFP in frame 1. (b) predicted locations of an SFP in frame 2 [5].

3.3.2. Modifying Location of SFPs as Needed

Due to the movement of objects in each frame and the boundary for speed of each object, some of the new SFPs can be uncontrollable, move unreasonably and in false locations in some frames. They will be allowed to move in legal fashion by defining some threshold values. so the velocity of SFPs is controlled and their location is modified.

$$SFP_j \ . p_l = \begin{cases} SFP_{selected\,(j)} \cdot p_l \ , \\ \quad if \ \left(SFP_{selected\,(j)} \cdot p_l \ ! \ SFP_j \ . p_l\right) \\ \qquad < v_{thr} \oplus coef \otimes v_{thr} \\ \quad and \ \left(SFP_{selected\,(j)} \cdot p_l \ ! \ SFP_j \ . p_l\right) \\ \qquad > v_{thr} \ ! \ coef \otimes v_{thr} \\ \\ SFP_j \ . p_l \oplus v_{thr} \quad otherwise \end{cases} \quad (7)$$

Vector v_{thr} is a threshold parameter for adjusting velocity of SFPs, $coef$ is the coefficient of velocity interval, and $SFP_{selected}(j)$ is the selected particle.

3.4. Identifying Outlier SFPs

Each SFP associated with an object O considers its relative location (rl) according to the left-top corner of the bounding box [5]. Given the relative location of an object SFP, the left-top corner of the object's bounding box is:

$$C_j = SFP_j \,.\, p_l \;!\; SFP_j \,.\, rl \tag{8}$$

$SFP_j \,.\, p_l$ is the location of the SFP and $SFP_j \,.\, rl$ is the relative location of it. According to Fig. 4. (a) all predictions must move to a point directly which is also the left-top corner of the bounding box in the frame.

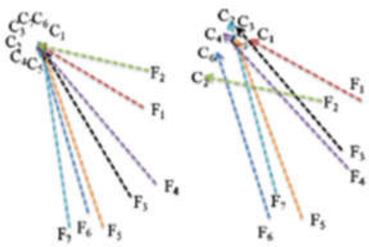

Fig. 4. Predicting the left-top corner of the bounding box by SFPs: (a) The ideal situation (b) The probable situation [5].

On the other hand, all SFPs may not be properly tracked depending on the filming conditions, the similarity to the background and the object's rotation, etc. In such a situation the prediction of all SFPs may not be the same. In Figure 4 (b), the predictions of all SFPs are close to each other except F_2. Hence prediction may lead to deviation or failure in tracking. For this reason, a timely prediction is right when it is consistent with other predictions of SFPs. Otherwise, it is treated as an outlier point. An SFP is specified as an outlier point if its prediction of the left-top corner, is more than twice of the standard deviation (assuming a confidence interval of 95%) of the average distribution C [5]. The outlier flag of an SFP is set this way:

$$SFP_j \,.\, outlier = \begin{cases} 0, & if \; (m \,!\, 2\alpha) \le C_j \le (m \oplus 2\alpha) \\ 1, & otherwise \end{cases} \tag{9}$$

α is the standard deviation of distribution C and m is the mean.

3.4.1. Updating the Bounding Box

The average predictions made by the correct tracked SFPs $(\mathrm{outlier} = \mathrm{overlapped} = 0)$ are calculated as the left-top corner of the bounding box [5].

$$BB \,.\, lt_j = SFP_j \,.\, p_l \;!\; SFP_j \,.\, rl \tag{10}$$

$SFP_j \,.\, rl$ is the relative location and $SFP_j \,.\, p_l$ is location of the tracked SFP:

$$BB \,.\, lt = \frac{1}{N} \sum_{j=1}^{N} BB \,.\, lt_j \tag{11}$$

N is the number of correct tracked SFPs.

Generally, the ratio of the bounding box size to the relative location of a tracked SFP is expected to be approximately the same in different frames. For a correct tracked SFP :

$$BB.\, s_j^t \;\%\; SFP_j \,.\, rl^t = BB.\, s_j^{t-1} \;\%\; SFP_j \,.\, rl^{t-1} \tag{12}$$

$BB.\, s^{t-1}$ is the bounding box size in frame $(t-1)$ and $SFP_j \,.\, rl^{\,t-1}$ is the relative location of the SFP in frame $(t-1)$. Using equation (12):

$$BB \,.\, s_j^t = BB \,.\, s^{t-1} \;\%\; SFP_j \,.\, rl^{t-1} \otimes SFP_j \,.\, rl^t \tag{13}$$

Bounding box size in frame t:

$$BB \,.\, s^t = \frac{1}{N} \sum_{j=1}^{N} BB \,.\, s_j^t \tag{14}$$

$BB \,.\, s^t$ is the bounding box size in frame t.

3.4.2. Modifying Outlier SFPs

The relative location of different SFPs in target objects may vary from one frame to another. Therefore after regulating the size of the bounding box, the locations of the detected SFPs remain unchanged, but the outlier SFP locations need to be revised. To modify the location of an outlier SFP based on the size of the bounding box and its previous relative location the following formula is used [5]:

$$SPF_j \, . \, p_l = \left(SFP_j \, . \, rl^{t-1} \right) \otimes \left(BB \, . \, s^t \right) \% \; BB \, . \, s^{t-1} \oplus BB \, . \, lt \qquad (15)$$

$SFP_j \, . \, p_l$ and $BB.\, lt$ are modified the location of the outlier SFP and left-top corner of the bounding box in the frame t and $SFP_j \, . \, rl^{t-1}$ is the relative location of the outlier SFP in the frame $(t-1)$.

An example of correcting the location of an outlier SFP is shown in Fig. 5. The SFP is near to the head (Fig. 5 (a)). In the next frame, the predicted spot of the particle with maximal weight is not backed by other SFPs and the particle is identified as an outlier point (Fig. 5 (b)). Hence, the location of this SFP is revised and corrected using its previous relative location (Fig. 5 (c)). After correction, the SFP is placed in a suitable location [5].

(a) (b) (c)

Fig. 5. Modifying an outlier SFP: (a) main location of an SFP. (b) predicted location of an SFP in the next frame. (c) location of the SFP after modifying [5].

3.4.3. Modifying Outlier SFPs

Overlapping is a very common phenomenon especially when we track several target objects. In such cases, a target object may be covered by partial or complete barriers or other objects. If two or more target objects look very near or overlap in a frame, some SFPs of a target object may overlap with another SFP of the same object or other. In this case, the feature of an SFP whose flag is overlapping will not be updated because updating may lead to an incorrect SFP tracking of another object. An example of an overlapping SFP is shown in Fig. 6. This figure displays an overlapping case beside the shoulder of the man with a black shirt (person 2), while the SFP basically belongs to person 1. If the SFP feature is updated at this location, the wrong SFP may be generated in subsequent frames and cause false results. An SFP is overlapping if it is located in more than one bounding box [5]:

$$SFP_j.Overlapped = \begin{cases} 1\,, if\ SFP_j.p_l \in B_p \cap B_q, p, q \in \mathbb{R} \\ \quad and\ p \neq q\ and\ p, q \leq m \\ 0\,,\ othrwise \end{cases} \qquad (16)$$

m is the number of objects and $\mathbb{R}$ is the set of real numbers.

Fig. 6. An overlapped SFP [5].

4. Experimental Results

In order to have a benchmark for evaluating the proposed algorithm and other algorithms the experiments were performed on the PETS2000 video sequence [11]. We have compared the proposed method with other powerful tracking algorithms using precision and recall to evaluate the performance [12]. It should be noted that these criteria are calculated in each frame of the video sequence and on the seven selected frames. The effectiveness of the proposed method have been compared with a number of competing methods including Kalman Filter with Occlusion Handling (KFOH)[13], Mean Shift (MS) [14] Corrected Background Weighted Histogram (CBWH) [15], and Multiple Object Tracking with Particle Occlusion Handling (MOPOH) [5] (Table 1).

$$precision = \frac{area\left(BB^{GT} \cap BB^{A\lg}\right)}{area\left(BB^{A\lg}\right)} \qquad (17)$$

$$recall = \frac{area\left(BB^{GT} \cap BB^{A\lg}\right)}{area\left(BB^{GT}\right)} \qquad (18)$$

In these two formulas, BB^{GT} and $BB^{\mathrm{A\,lg}}$ are bounding boxes provided by the ground truth and the algorithm respectively. $area$ shows the number of pixels intrant the bounding box [5].

Table. 1. Methods used in comparison

#	Method	Full Name	Reference
1	KFOH	Kalman Filter with Occlusion Handling	[13]
2	MS	Mean Shift	[14]
3	CBWH	Corrected Background Weighted Histogram	[15]
4	MOPOH	Tracking with Particle Occlusion Handling	[5]

4.1. Evaluation on PETS2000 Database

The brightness of the images is lower than other standard sequences in this database and it includes two moving objects. The images also display a number of cars of the same color that cause problems for most of the robust tracking algorithms. KFOH has encountered difficulties during the occlusion in frames 8 to 18 (Fig. 7 (a)). However, MS successfully isolated two individuals in frame 18 to 25, despite the fact that the stability of bounding boxes is not preserved during the occlusion due to the large similarity between the car and the woman (Fig. 7 (b)). The CBWH also fails to track a man who wears a black coat (Figure 7 (c)), and finds the common characteristics of the other person and the car. In frames 8 to 28 the CBWH cannot keep the bounding box for the relevant person. Some SFPs of the target object towards the car in frames eight to eighteen have been mistaken by MOPOH (Figure 7 (d)).

Table 2 and Table 3 show the accuracy of the method on both objects. Although the KFOH recall remains high, the precision of the frames 10 to 18 is low because it creates a big bounding box during this period. The recall and precision of MS and CBWH in the frame 15 is very low for the reason that the bounding box is generated in wrong places. Accuracy and recall are gently decreasing for CBWH, as soon as it fails to track the man (second object) after several frames. The proposed method has a higher accuracy and recall than compared methods.

Table. 2. The precision of methods for two objects in PETS2000 sequence

Frame	KFOH	MS	CBWH	MOPOH	Proposed method
2	0.8000	0.7630	0.7490	0.7782	0.8345
8	0.7651	0.6620	0.6669	0.7445	0.7796
10	0.7389	0.7200	0.6575	0.7795	0.7974
12	0.7417	0.7069	0.6324	0.7799	0.7949
15	0.7028	0.2466	0.0467	0.7470	0.8183
18	0.6834	0.4434	0.1376	0.7349	0.7551
25	0.6537	0.5720	0.2562	0.7438	0.7979
Average	0.7265	0.5877	0.4495	0.7582	0.7968

Table. 3. Recall of methods for two objects in PETS2000 sequence

Frame	KFOH	MS	CBWH	MOPOH	Proposed method
2	1	0.8900	0.8544	0.8724	0.9100
8	0.8987	0.8036	0.7886	0.8821	0.9442
10	0.5374	0.8407	0.7621	0.8024	0.8745
12	0.5398	0.8452	0.7526	0.8215	0.7631
15	0.3833	0.4144	0.2256	0.7281	0.9292
18	0.3230	0.5526	0.3512	0.7077	0.9622
25	0.8814	0.7529	0.4457	0.7711	0.8885
Average	0.5619	0.7285	0.5972	0.7979	0.8960

5. Conclusion

One of the most challenging issues in computer vision is the tracking of moving objects. Occlusion is considered as one of these challenging issues in object tracking, which results in a dramatic decrease in tracking accuracy and in the video sequence. The proposed approach for tracking under partial occlusion includes four phases. We extract the background and draw bounding boxes around the objects in the first phase or preprocessing. In the second phase, the feature extraction is done using a corner detection algorithm to explore the salient feature points or SFPs of the objects and their features are calculated. In the third phase, the particle filter is used, so the SFPs are replaced by particles with the highest weight. The final phase is to update and modify the points where new points of the particles with maximal weights are modified and updated if needed (points with occlusion, outlier points and some points with irrational speed). Then, after updating the bounding box, the outlier points are modified as well as the points with occlusion. All of the different evaluation criteria indicate the ability and productiveness of the suggested approach to handle the tracking problem under partial occlusion conditions.

Fig. 7. Tracking results on seven frames (second, eighth, twelfth, fifteenth, eighteenth and twenty-fifth) of PETS2000 video sequence. (a)KFOH, (b) MS, (c) CBWH, (d) MOPOH, (e) the suggested method.

References

[1] Yilmaz, A.; Javed, O.; Shah, M., "Object tracking: A survey", Acm computing surveys (CSUR), vol. 38, p. 13 (2006).

[2] Watanabe, G.; Fukui, S.; Iwahori, Y.; Bhuyan, M K.; Woodham, R J.; Adachi, Y.; "Tracking Method in Consideration of Existence of Similar Object around Target Object", Procedia Computer Science, vol. 22, pp. 366-374 (2013).

[3] Cancela, B.; Ortega, M.; Fernandez, A.; Penedo, M G., "Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios", Expert Systems with Applications, vol. 40, pp. 1116-1131 (2013).

[4] Liu, H.; Sun, F., "Efficient visual tracking using particle filter with incremental likelihood calculation", Information Sciences, vol. 195, pp. 141-153 (2012).

[5] Naushad Ali, M. M.; Abdullah-Al-Wadud, M.; Lee, S. L., "Multiple object tracking with partial occlusion handling using salient feature points", Information Sciences, vol. 278, pp. 448-465 (2014).

[6] Harris, C.; Stephens, M., "A combined corner and edge detector", Alvey vision conference , pp. 10-5244 (1988).

[7] Dalal, N.; Triggs, B., "Histograms of oriented gradients for human detection", IEEE, Computer Vision and Pattern Recognition (CVPR). pp. 886-893 (2005).

[8] Xu, R.; Nikouei, S. Y.; Chen, Y.; Polunchenko, A.; Song, S.; Deng, C.; Faughnan, T. R., "Real-time human objects tracking for smart

surveillance at the edge", IEEE, International Conference on Communications (ICC), pp. 1-6 (2018).

[9] Zhou, S.; Ke, M.; Qiu, J.; Wang, J., "A Survey of Multi-object Video Tracking Algorithms", International Conference on Applications and Techniques in Cyber Security and Intelligence. pp. 351-369 (2018).

[10] Xiao, J.; Oussalah, M., "On the use of contextual information for robust colour-based particle filter tracking", Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA) , pp. 1-6 (2018).

[11] f. f. p. r. a. u. p. PETS2000/.

[12] Sobti, A.; Arora, C.; Balakrishnan, M., "Object detection in real-time systems: Going beyond precision", IEEE. Winter Conference on Applications of Computer Vision (WACV), pp. 1020-1028 (2018).

[13] Li, X.; Wang, K.; Wang, W.; Li, Y., "A multiple object tracking method using Kalman filter", IEEE. International Conference on Information and Automation (ICIA). pp. 1862-1866 (2010).

[14] Comanici, D. u.; Ramesh, V.; Meer, P., "Kernel-based object tracking", IEEE. Transactions on pattern analysis and machine intelligence, vol. 25, pp. 564-577 (2003).

[15] Ning, J.; Zhang, L.; Zhang, D.; Wu, C., "Robust mean-shift tracking with corrected background-weighted histogram", IET computer vision, vol. 6, pp. 62-69 (2012).