Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4039002, 9 pages
https://doi.org/10.1155/2022/4039002

Research Article

Adaptive Tracking Algorithm for Multiperson Motion Targets Based on Local Feature Similarity

Chang Xiankun,[1] Liu Mingliang,[2] and Wang TianMing[3]

[1]*Department of Physical Education, Shanghai University of Engineering Science, Shanghai, China 201620*
[2]*Department of Physical Education and Aesthetic Education, University of International Relations, Beijing, China*
[3]*China Committee on Care for the Next Generation Health and Sports Development Center, Beijing, China*

Correspondence should be addressed to Liu Mingliang; sconquer@sues.edu.cn

Received 20 February 2022; Accepted 9 March 2022; Published 16 April 2022

Academic Editor: Kalidoss Rajakani

The traditional multitarget tracking method has many points of concern. In view of the problems of high target recognition switching rate and high target trajectory false alarm rate in complex scenes and the problem of target loss caused by video tracking in existing video tracking algorithms, a new method based on video tracking is proposed: a multitarget tracking algorithm for local feature similarity, a strong target motion maneuvering or rapid deformation of asymmetric rigid targets. The algorithm builds a depth metric model, which can predict and track the temporal features of the target trajectory frame appearance features and motion features at the same time, which makes the extracted target features more discriminative and reduces the target recognition switching rate. At the same time, the adaptive model tracking algorithm can adjust the model in real time according to the clarity of the target area, effectively ensure the accuracy of the target tracking model, effectively aggregate the characteristics of the trajectory frame, and reduce the false alarm rate. The experimental results show that combining the multiperson moving target adaptive tracking algorithm based on local feature similarity into the DSST model can improve the average accuracy and success rate of the DSST model and has good stability.

1. Introduction

Identifying and monitoring multiple target items such as pedestrians and automobiles is accomplished through the analysis of video. In practical applications like video surveillance, autonomous driving, motion identification, and crowd behavior analysis, multitarget tracking algorithms are used to track many targets simultaneously. Presently, multitarget tracking is a tough visual task, with the primary issue being easy to lose track of the tracked object or switch the target identity when tracking numerous targets at the same time, due to occlusion and a high degree of resemblance in their appearance. The success of deep learning in computer vision tasks such as picture classification and target recognition has fueled the advancement of multitarget tracking technology in recent years, and multitarget tracking algorithms integrated with deep neural networks have emerged as a popular research area [1–7].

Detection-based multitarget tracking frameworks are comprised of the following processes: target detection, feature extraction from the target detection frame, similarity computation between the target detection frame and the trajectory frame, and data association. The target detection algorithm is responsible for the majority of the work in the detection phase, and the present research on multitarget tracking algorithms is mostly concerned with the latter two things.

Visual target tracking has long been a popular research topic in the field of computer vision, and as technology has progressed, target tracking technology has become increasingly popular in a variety of fields, including unmanned aerial vehicles, intelligent traffic systems, intelligent surveillance systems, virtual reality, and other applications [8, 9]. When the target is hidden during the tracking process, the light conditions in the environment change, the target scale changes, and so on, this might result in a poor eye-tracking effect. A large number of academics have

undertaken some research on tracking algorithms in order to solve this type of challenge [10–15].

The kernel correlation filter (KCF) tracking algorithm has been proposed by some researchers, which converts the matrix operation into an intervector dot product operation by extracting the image direction HOG features and exploiting the fact that a circular matrix can be diagonalized in the frequency domain to speed up the operation [16, 17]. However, the method has a drawback: it cannot handle target scale changes and is unable to perform scale adaptation. The adaptive color attribute tracking approach translates the characteristics of RGB space into an 11-dimensional color space, which increases the abundance of the algorithm's input features and enhances the tracking accuracy by making the algorithm's input features more numerous [18]. The target is taught to compensate for the impacts of scale transformation and illumination on the tracking effect by learning to track in different environments [19]. The scale adaptive correlation filter SAMF algorithm, which employs the scale pooling approach to achieve adaptive tracking of the target, is described in detail below. Some researchers have improved the KCF algorithm by including an adaptive Gaussian window function and scale estimation. Their findings have been promising. When confronted with difficulties like as background clutter, rapid target motion, light changes, occlusion, and size changes, the tracking accuracy of the algorithms listed above will deteriorate to varying degrees, and in some cases, the target will be lost entirely [20–24].

With the advancement of deep learning technology, it is possible to mine the deep semantic information contained in images and apply it in sectors such as target tracking and image segmentation, among others [25]. The HCF tracking algorithm, which is based on the kernel correlation filter tracking algorithm, selects multilayer depth features rather than HOG features and uses shallow network features for tracking and localization, which improves the tracking accuracy. The HCF tracking algorithm is based on the kernel correlation filter tracking algorithm [26, 27]. The Siam R-CNN algorithm presents a new method of mining instances. The C-COT algorithm is an improvement on deep learning and correlation filtering algorithms that addresses the problem of algorithm training in the continuous space domain by using cubic interpolation and a Hessian matrix to handle the problem of algorithm training in the continuous space domain [28].

The circular structure kernel (CSK) algorithm, the kernel correlation filtering (KCF) algorithm, the discriminative scale space tracking (DSST) algorithm, the edge target tracking (LMCF) algorithm based on circular feature mapping, and the long-time target tracking (LCT) algorithm, among others, are the most widely used correlation filtering algorithms at this time. As part of its eye-tracking accuracy enhancements, the CSK algorithm incorporates a kernel-based least-sum-of-squares filter (MOSSE), which is based on the Ridge regression approach based on the kernel function. This helps to maintain the algorithm's computing performance while improving accuracy. CSK is used to create multichannel features, which are then used by KCF in order to improve the trackability and tracking accuracy of the algorithm at the feature extraction level. The DSST method incorporates scale tracking into the correlation filtering algorithm in order to address the issue of the impact of targets with large-scale changes on the tracking results of the correlation filtering algorithm [29]. The LMCF algorithm approaches model updating from the standpoint of tracking accuracy, and the average peak correlation energy (APCE) metric and feature maximum response value are combined to determine tracking accuracy and, consequently, whether or not to update the model is necessary. On the basis of DSST, the LCT method integrates online neighborhood algorithm (KNN) detection, which significantly increases the stability of the system for long-term tracking applications [30]. The tracking effects of the algorithms described above are good in the case of weak maneuverability and modest deformation of the target in the video, and they can increase the tracking accuracy of the target to a certain level in this scenario. However, if the target is more mobile or changes rapidly or the image is obscured, the target is easily lost and cannot be recovered, resulting in low tracking accuracy and long-term tracking [31].

A multiperson motion target adaptive tracking algorithm based on local feature similarity is proposed in this paper to address the problems mentioned above. This algorithm can effectively improve tracking accuracy in the case of a strong target motion maneuver or rapid deformation of asymmetric rigid targets. Furthermore, the target's appearance features and motion features are extracted uniformly in the depth metric network in order to learn their temporal correlation, which improves the target's discriminative property and minimizes the target's ID switching rate. A network is also trained to learn the probability value of correct matching of different time sequence history trajectory frames of the target, to suppress false detection in the target trajectory and the influence of low-quality target frames on the overall features of the target, to reduce the false alarm rate, and to solve the problem that a target cannot be retrieved again after it has been lost, all at the same time.

The research contributions of the paper are as follows:

(1) This paper proposes a multitarget tracking algorithm based on local feature similarity

(2) The algorithm proposed in this paper builds a depth metric model, which can simultaneously predict the temporal features of the appearance features and motion features of the tracking target trajectory frame

(3) The proposed adaptive model tracking algorithm can adjust the model in real time according to the clarity of the target area, effectively ensure the accuracy of the target tracking model, effectively aggregate the characteristics of the trajectory frame, and reduce the false alarm rate

2. Related Method

2.1. Correlation Filtering Tracking. Correlation filtering was first applied in signal processing as a method to describe the correlation between 2 signals. For 2 signals f and g, their correlation in the time domain is in the continuous and discrete forms as follows:

$$(f \otimes g)\tau = \int_{-\infty}^{+\infty} f*(t)g(t+\tau)dt,$$
$$(f \otimes g)n = \int_{-\infty}^{+\infty} f*(m)g(m+\tau)dm, \quad (1)$$

where $f*(t)$ is the complex covariance of $f(t)$, t and τ are the independent variables and variations in continuous time, and m and n are the independent variables and variations in discrete time, respectively. The correlation calculation is performed for $f(t)$ and $f(m)$. The moment corresponding to the maximum response value is the moment when the two signals are closest.

According to the fundamental concept of the correlation filtering algorithm, the tracking problem of the target position in an image may be characterized as follows: Figure 1 depicts the process of tracking the target position in an image. As a result, it can be determined that the correlation filtering algorithm requires the extraction of a target image filter template h_i in the initial frame and the calculation of the target image filter template h_i with the input image f_i in order to produce the response matrix g_i. In this case, the subscript i represents the pixel ordinal number. The greater the distance between the point and the target, the greater the response value. When the tracking is steady, the response matrix g_i calculated from the tracking results will have a good single peak; however, when the tracking is unstable, the response matrix g_i will have several peaks with low peaks.

The calculation process of the correlation filter tracking algorithm is shown in Figure 2, and it can be found that the whole calculation process is mainly divided into three parts: feature extraction, online matching, and model updating. First, the features are extracted from the target of the first frame of the image sequence, including grayscale features, color features, HOG features, and depth features in the deep learning algorithm; then, the extracted feature model is matched online with the region to be matched in the next frame to get the response matrix of the next image. After matching each frame, the obtained feature model is updated according to a certain learning rate to quadruple-match the next frame with the learned new model, and the model update can be expressed as

$$\hat{X}^p = (1-\gamma)\hat{X}^{p-1} + \gamma\hat{X}, \quad (2)$$

where $\hat{X}$ is the extracted feature model and γ is the learning rate.

2.2. KCF. The full name of the KCF algorithm is kernelized correlation filters, which is a fully automatic tracking algorithm. The KCF algorithm consists of several steps: first, use a circular matrix to sample the positive and negative samples of the tracking area, extract the sample HOG features, and train the filter. Secondly, the similarity between the target area and the candidate area is calculated by using the polynomial kernel function, the response graph is derived, and the peak position in the response graph represents the tracking target position; thirdly, the tracking target position is calculated using the polynomial kernel function.

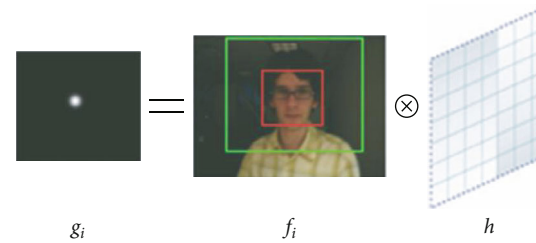

Figure 1: Structure of linear CRF.

Let the training sample be x_i, the corresponding label be y_i, and linear classifier be $f(x) = W^T x_i$; the purpose of the training classifier is to find the best filter template coefficient W by sample x_i, so that $f(x_i)$ and label y_i squared error sum are minimum; the loss function of the classifier is shown in the following:

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2, \quad (3)$$

in which λ is a regularized item.

By introducing the mapping function $\varphi(x)$, the data is mapped to the high-dimensional space, at which point W can be expressed as a linear combination of $\varphi(x_i)$, as shown in the following equation, and can be solved using the kernel function.

$$W = \sum_i \alpha_i \varphi(x_i). \quad (4)$$

Then we have

$$\alpha = (k + \lambda I)^{-1} y, \quad (5)$$

in which I is the unit matrix, y_i is the sample label, and k is the kernel function.

By using the circular matrix and Fourier transform operation, we have

$$\alpha = F^{-1}\left|\frac{F(y)}{F(k)+\lambda}\right|, \quad (6)$$

in which F^{-1} denotes the Fourier transform operation.

When dealing with targets that move a lot or deform a lot in the video, it is critical to have a robust adaptive model updating process. As a result, the learning rate of the model is determined by the relative sharpness of the candidate region in the next frame, which is determined by the learning rate algorithm. *Tenengrad*, an image gradient-based function and a generally used image sharpness assessment function, is used to judge the sharpness of the candidate region in order to keep the algorithm's speed as low as possible. Image processing professionals often feel that clearer image pixels have sharper edges, for example, greater gradient function values, and blurrier image pixels have less sharp

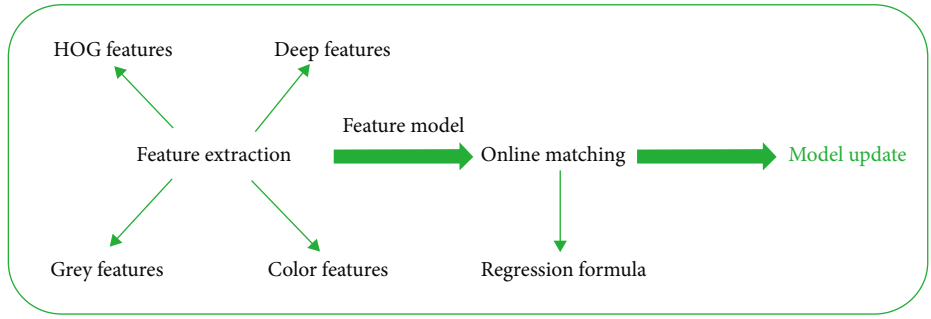

Figure 2: Structure of correlation filter tracking algorithm.

edges. It is necessary to utilize the *Sobel* operator to extract gradient values in both the horizontal and vertical directions when using the *Tenengrad* function, and the following are the specific procedures to be taken.

(1) Let the *Sobel* convolution kernel be G_x and G_y; then the gradient of image I at the point (x, y) can be expressed as

$$S(x,y) = \sqrt{G_x * I(x,y) + G_y * I(x,y)}. \tag{7}$$

And the *Tenengrad* function is

$$X_{Tenengrad} = \frac{1}{n}\sum_x \sum_y S(x.y)^2, \tag{8}$$

in which n is the total number of pixels in the evaluation region. A larger value of the *Tenengrad* function indicates a higher sharpness of the evaluation region. Conversely, it indicates a lower sharpness of the evaluation region. First, a basic learning rate L_{base} and a basic sharpness C_{base} are fixed, where the basic sharpness C_{base} is obtained from the sharpness evaluation value of the first frame. Then, the ratio of the basic sharpness C_{base} to the sharpness of the current frame is calculated in the remaining frames, and the product of this ratio and the basic learning rate is defined as the learning rate. Finally, the target loss caused by inaccurate feature extraction when the motion target is too blurred is fully considered, and the sharpness threshold T_C. If the sharpness is lower than the threshold T_C, then the learning rate is immediately adjusted to 0, which can be expressed as

$$\begin{cases} I = L_{base} \times \dfrac{C_{base}}{c}, c > T_C \\ I = 0, c < T_C \end{cases}, \tag{9}$$

in which I is the updated learning rate and C is the clarity value at the current moment.

Since the continuous movement of the target will lead to changes in the appearance model, which directly affects the tracking effect, it is necessary to continuously update the filter and appearance model:

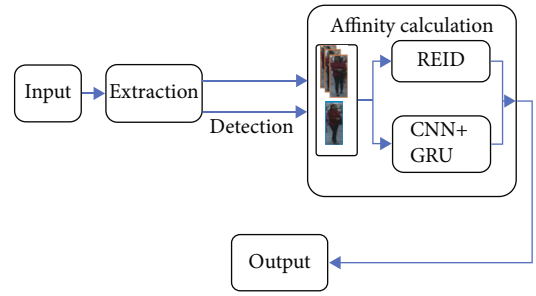

Figure 3: The structure of our algorithm.

$$\begin{aligned} x_i &= (1-\eta)x_{i-1} + \eta x_i, \\ \alpha_i &= (1-\eta)x_{i-1} + \eta \alpha_i, \end{aligned} \tag{10}$$

in which, x_i is the appearance model, α_i is the classifier coefficient, and α_i is the learning rate.

3. Adaptive Tracking Algorithm for Multiperson Motion Targets Based on Local Feature Similarity

The structure of the adaptive tracking algorithm for multiperson motion targets based on local feature similarity is shown in Figure 3.

Specifically, this multitarget tracking framework is composed of the following four components:

(1) Extraction of the target frame. It is retrieved from the current frame of video based on its target detection algorithm, whereas it is computed from its past video frames to produce a track frame, which is the target track frame derived from its historical video frames

(2) The calculation of similarity. A metric network is utilized to determine the degree of similarity between the target detection frame and the trajectory frame, and the result is displayed. This network uses the CNN to extract the depth features of the target frame, and then two GRUs are used to learn the temporal correlation of the appearance and motion features of the target historical track frame and the probability value of correct matching for each historical track frame saved

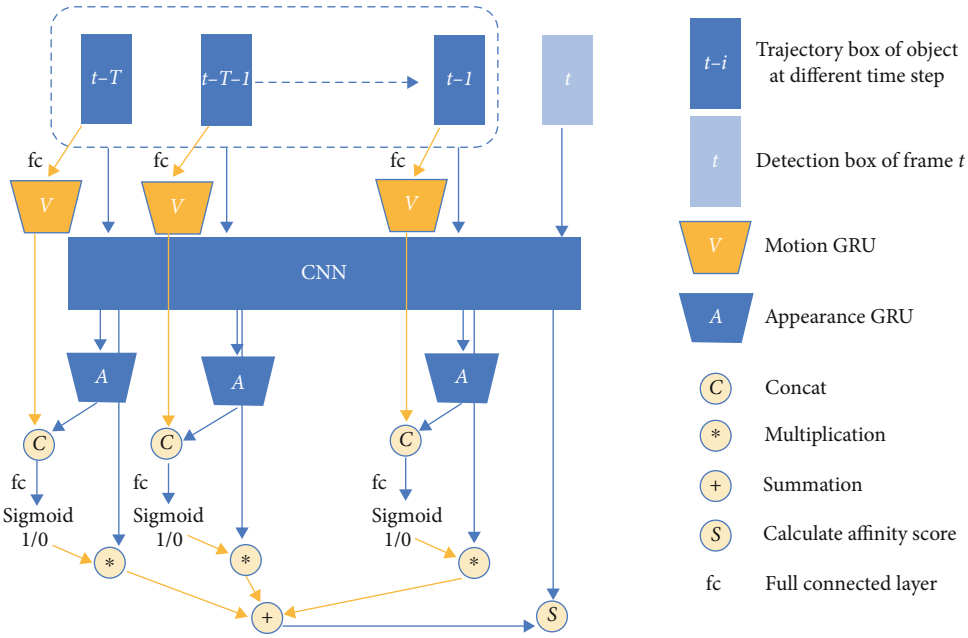

Figure 4: The structure of CNN-GRU.

by the target, to aggregate the appearance features of the target track frame in different time sequences, and then the CNN-GRU network outputs the similarity between the f and the target track frame, as shown in Figure 1. In the following step, the CNN-GRU network determines if the features of the target trajectory frame and the detection frame are identical. For the target detection frame and trajectory frame, a deep learning-based Reid network is used to extract the appearance features from their respective frames and to determine the similarity between them. The network structure of CNN-GRU is shown in Figure 4

(3) This paper addresses the issues of tracking accuracy and robustness of the relevant filter tracking algorithm in complex environments. Because the KCF target tracking algorithm using HOG features may lead to tracking drift or even tracking failure due to the single extracted feature, this paper replaces the single feature of the original KCF algorithm with a weighted fusion of FHOG features and depth features to complete the accurate localization of the target and completes the est. tracking accuracy

(4) The association of data. The similarity between the output of the Reid network and CNN-GRU is combined to obtain the matching correlation matrix of detection frames and trajectory frames, and the matching results of all detection frames and target trajectory frames of the current video frame are finally obtained by the Hungarian matching algorithm, which is based on the similarity between the output of the Reid network and CNN-GRU

Table 1: Comparison of tracking results with different λ_1, λ_2, and λ_3.

Weight			Accuracy	Success rate
λ_1	λ_2	λ_3		
1	0.2	0.5	0.88	0.74
1	0.4	0.5	0.85	0.71
1	0.6	0.5	0.91	0.79
1	0.8	0.5	0.81	0.62
1	1	0.5	0.86	0.73

By directly training the similarity between detection frames and trajectory frames and by adaptively combining appearance features and motion features in this metric network, the framework is able to cope with complex scene changes of multitarget tracking and cope with complex scene changes of multitarget tracking. The deep metric network is trained to understand the temporal connection of appearance and motion attributes of historical target trajectory frames in distinct time sequences, which reduces the ID switching rate of targets. Learning the probability value of correct matching of previous track frames from distinct time sequences saved by each target helps to limit the number of false alarms generated. As part of this process, the similarity of the depth metric network's output and the similarity of appearance features extracted by the pedestrian reidentification network are combined in order to obtain a final matching result between the detection frame and the trajectory frame, which further reduces the target identification switching rate.

Conv2-1 and Conv4-1 in ResNet-50 are used to extract the depth features of the tracking target region, and the FHOG features of the target region are also extracted. The

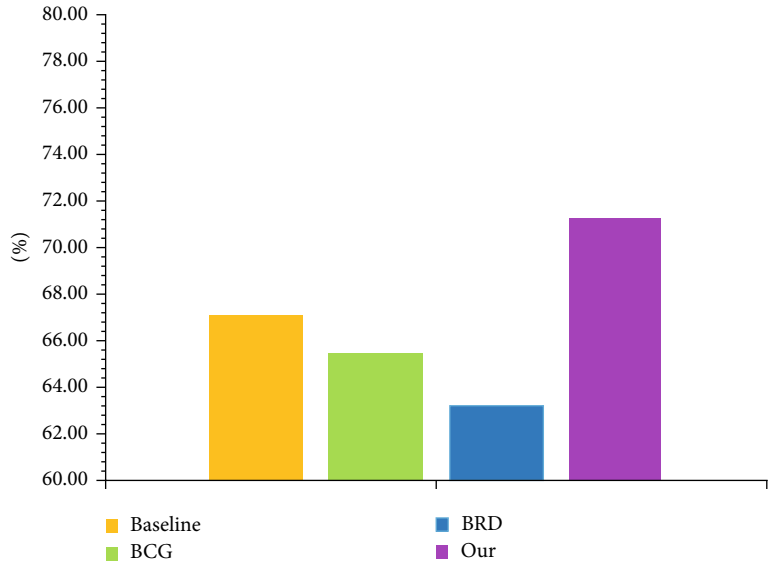

Figure 5: Comparison of the IDF1 value among different algorithms.

response of each feature in the frequency domain is calculated as shown in the following equations.

$$f_{Conv2-1}(z) = F^{-1}[F(k_{zxConv2-1}) \odot F(\alpha_{Conv2-1})], \quad (11)$$

$$f_{Conv4-1}(z) = F^{-1}[F(k_{zxConv4-1}) \odot F(\alpha_{Conv4-1})], \quad (12)$$

$$f_{FHOG}(z) = F^{-1}[F(k_{zxFHOG}) \odot F(\alpha_{FHOG})], \quad (13)$$

in which $f_{Conv2-1}(z)$, $f_{Conv4-1}(z)$, and $f_{FHOG}(z)$ represent the response maps corresponding to the Conv2-1 feature, Conv4-1 feature, and FHOG feature of the sample z of the window to be detected, respectively. α represents the solution of the classifier corresponding to each feature, and k_{zx} represents the kernel correlation between the feature vector x corresponding to the training sample and the sample z of the candidate region in the next frame.

The 3 feature response maps are weighted and fused as shown in the following

$$f(z) = \lambda_1 f_{Conv2-1}(z) + \lambda_2 f_{Conv4-1}(z) + \lambda_3 f_{FHOG}(z), \quad (14)$$

in which λ_1, λ_2, and λ_3 are feature fusion coefficients; it was found during the experiments that the 3 response maps were fused with different weights with different effects, and the comparison of tracking results under different weights is shown in Table 1. The improved algorithm fusion weight into $\lambda_1 = 1$, $\lambda_2 = 0.6$, and $\lambda_3 = 0.5$ is derived through extensive experiments. $f(z)$ is the final fused response map, and the peak position is the tracking target center position.

It employs the YOLOv4 detection algorithm, which has a fair balance between speed and accuracy at this point in the development process. Because YOLOv4 is a deep learning-based target detection method, it is necessary to train the algorithm on the category of the tracked target before it can be used; otherwise, the system will not be able to recognize and track unknown targets. The target bounding box

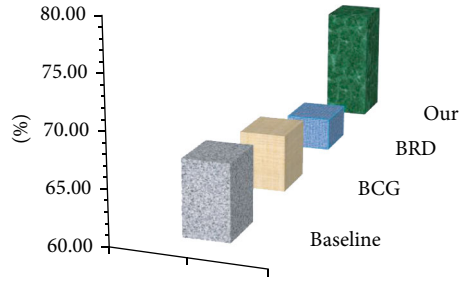

Figure 6: Comparison of the IDF1 value among different algorithms.

detected by the YOLOv4 algorithm will have a confidence score, and if the confidence score is greater than a predetermined threshold, the program automatically determines that the detection was successful, and the algorithm moves on to the tracking phase, with the detected bounding box serving as the initial frame for the tracking phase. Whenever the greatest response value of multiple consecutive frames is less than the preset threshold, it is possible to conclude that the tracking has drifted at that point; in this case, the recheck mode is activated, and the detection process is entered once more.

4. Experiment Results

The experiments focus on the tracking results of multiplayer motion targets, where the comparison experiments on the MOT17 dataset use the open target detection results provided by this dataset. The detection results of the tracker detection part of the validation experiments on the MOT16 dataset use the results from the literature [21].

To verify the effectiveness of the adaptive tracking algorithm for multiperson motion targets based on local feature similarity, validation experiments are first conducted on the

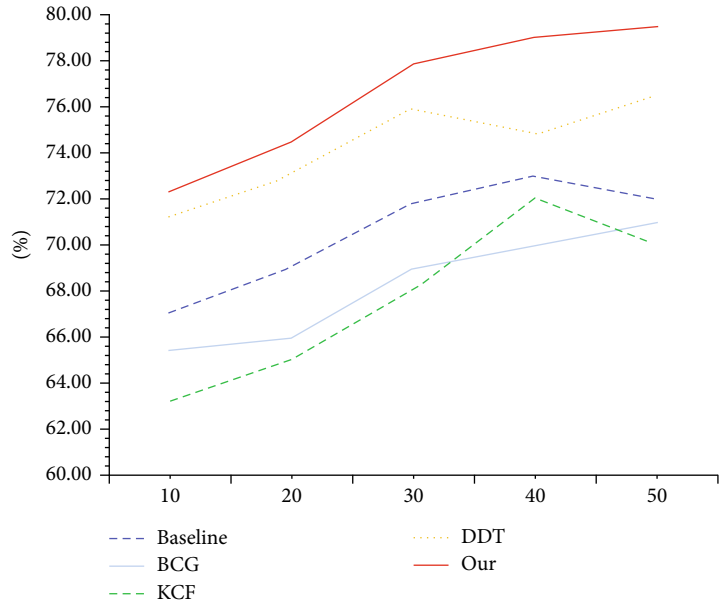

Figure 7: Comparison of algorithms with different error thresholds.

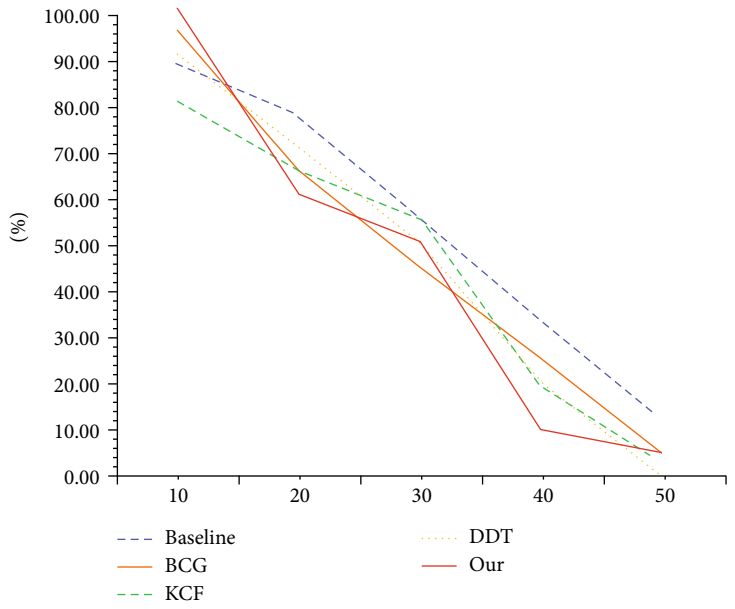

Figure 8: Comparison of algorithms with different overlap thresholds.

MOT16 training set. Let the baseline model of the tracker consists of Kalman filter + IOU + Hungarian matching algorithm, named baseline. The multitarget tracker consisting of the baseline model + metric network is named BCG. The multitarget tracker consisting of the baseline model + pedestrian reidentification network is named BRD.

Figures 5 and 6 represent the comparison results of the algorithm in this paper and the comparison algorithm on the two metrics of IDF1 and IDs, respectively. It can be seen that the algorithm in this paper significantly outperforms the comparison algorithm.

Both Figures 7 and 8 depict the OPE test curves for all of the targets in the OTB100 dataset that were subjected to fast motion, motion blur, and fast deformation rapid deformation. In this study, it is discovered that the average accuracy and success rate of the algorithm are greater than the average accuracy and success rate of the other algorithms on the rapid motion target frame sequences. This suggests that the approach employed by this algorithm can, to a certain extent, mitigate the target loss induced by a powerful target motion maneuver or by rapid deformation of asymmetric rigid targets.

It can be seen from Figures 7 and 8 that the threshold of the algorithm proposed in this paper is higher and more reasonable than other algorithms, and it has more advantages under the same conditions.

5. Conclusion

This paper proposes an adaptive tracking algorithm for multiperson moving targets based on local feature similarity, which solves the problem of target loss caused by strong target motion maneuvering or rapid deformation of asymmetric rigid targets to a certain extent. This method uniformly extracts the appearance features and motion features of targets in the metric network and learns their temporal correlations, so that the targets have better discriminativeness, thereby reducing the ID switching rate. The experimental results show that the algorithm in this paper is obviously better than the comparison algorithm. The framework proposed in this paper can effectively reduce the ID switching rate and false alarm rate and improve the tracking accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflict of interest.

References

[1] J. Zhang, L. L. Presti, and S. Sclaroff, "Online multi-person tracking by tracker hierarchy," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pp. 379–385, IEEE, 2012.

[2] Q. Zhou, B. Zhong, Y. Zhang, J. Li, and Y. Fu, "Deep alignment network based multi-person tracking with occlusion and motion reasoning," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1183–1194, 2019.

[3] K. Bernardin and R. Stiefelhagen, "Audio-visual multi-person tracking and identification for smart environments," in *Proceedings of the 15th ACM international conference on Multimedia*, pp. 661–670, 2007.

[4] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6036–6046, 2018.

[5] M. Luber, L. Spinello, and K. O. Arras, "People tracking in rgb-d data with on-line boosted target models," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3844–3849, IEEE, 2011.

[6] H. Kieritz, S. Becker, W. Hübner, and M. Arens, "Online multi-person tracking using integral channel features," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 122–130, IEEE, 2016.

[7] Y. Zhang, "Detection and tracking of human motion targets in video images based on camshift algorithms," *IEEE Sensors Journal*, vol. 20, no. 20, pp. 11887–11893, 2019.

[8] S. Zhang, Y. Gong, J. B. Huang et al., "Tracking persons-of-interest via adaptive discriminative features," in *European Conference on Computer Vision*, pp. 415–433, Springer, Cham, 2016.

[9] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1515–1522, IEEE, 2009.

[10] B. Li, C. Yang, Q. Zhang, and G. Xu, "Condensation-based multi-person detection and tracking with HOG and LBP," in *2014 IEEE International Conference on Information and Automation (ICIA)*, pp. 267–272, IEEE, 2014.

[11] L. Ma, S. Tang, M. J. Black, and L. Van Gool, "Customized multi-person tracker," in *Asian Conference on Computer Vision*, pp. 612–628, Springer, Cham, 2018.

[12] M. Yang and Y. Jia, "Temporal dynamic appearance modeling for online multi-person tracking," *Computer Vision and Image Understanding*, vol. 153, pp. 16–28, 2016.

[13] J. Ju, D. Kim, B. Ku, D. K. Han, and H. Ko, "Online multi-person tracking with two-stage data association and online appearance model learning," *IET Computer Vision*, vol. 11, no. 1, pp. 87–95, 2017.

[14] A. Heili, A. López-Méndez, and J. M. Odobez, "Exploiting long-term connectivity and visual motion in CRF-based multi-person tracking," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3040–3056, 2014.

[15] N. Narayan, N. Sankaran, D. Arpit, K. Dantu, S. Setlur, and V. Govindaraju, "Person re-identification for improved multi-person multi-camera tracking by continuous entity association," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 64–70, 2017.

[16] A. López, C. Canton-Ferrer, and J. R. Casas, "Multi-person 3D tracking with particle filters on voxels," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*IEEE.

[17] S. Park and M. M. Trivedi, "Multi-person interaction and activity analysis: a synergistic track-and body-level analysis framework," *Machine Vision and Applications*, vol. 18, no. 3, pp. 151–166, 2007.

[18] K. Bernardin, T. Gehrig, and R. Stiefelhagen, "Multi-level particle filter fusion of features and cues for audio-visual person tracking," in *Multimodal Technologies for Perception of Humans*, pp. 70–81, Springer, Berlin, Heidelberg, 2007.

[19] S. Gao, Q. Ye, L. Liu, A. Kuijper, and X. Ji, "A graphical social topology model for RGB-D multi-person tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4305–4320, 2021.

[20] S. Jia, S. Wang, L. Wang, and X. Li, "Human tracking system based on adaptive multi-feature mean-shift for robot under the double-layer locating mechanism," *Advanced Robotics*, vol. 28, no. 24, pp. 1653–1664, 2014.

[21] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," in *The 2010 IEEE international conference on information and automation*, pp. 1862–1866, IEEE, 2010.

[22] H. Wu, Y. Hu, K. Wang, H. Li, L. Nie, and H. Cheng, "Instance-aware representation learning and association for online multi-person tracking," *Pattern Recognition*, vol. 94, pp. 25–34, 2019.

[23] J. C. Jinhua Wang and Y. Y. Di Wu, "An object tracking algorithm based on the" current" statistical model and the multi-feature fusion," *Journal of Software*, vol. 7, no. 9, 2012.

[24] G. Wang, Y. Wang, H. Zhang, R. Gu, and J. N. Hwang, "Exploit the connectivity: multi-object tracking with trackletnet," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 482–490, 2019.

[25] P. Huang, S. Han, J. Zhao et al., "Refinements in motion and appearance for online multi-object tracking," 2020, http://arxiv.org/abs/2003.07177.

[26] D. Stadler and J. Beyerer, "Modelling ambiguous assignments for multi-person tracking in crowds," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 133–142, 2022.

[27] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 1918–1925, IEEE, 2012.

[28] F. Abdullah, Y. Y. Ghadi, M. Gochoo, A. Jalal, and K. Kim, "Multi-person tracking and crowd behavior detection via particles gradient motion descriptor and improved entropy classifier," *Entropy*, vol. 23, no. 5, p. 628, 2021.

[29] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous association graph fusion for target association in multiple object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3269–3280, 2019.

[30] M. Harville, "Stereo person tracking with adaptive plan-view templates of height and occupancy statistics," *Image and Vision Computing*, vol. 22, no. 2, pp. 127–142, 2004.

[31] T. Linder and K. O. Arras, "Multi-model hypothesis tracking of groups of people in RGB-D data," in *17th international conference on information FUSION (FUSION)*, pp. 1–7, IEEE, 2014.