Hindawi
Complexity
Volume 2021, Article ID 6678596, 11 pages
https://doi.org/10.1155/2021/6678596

Research Article

A Specific Algorithm Based on Motion Direction Prediction

Zhesen Chu[1] **and Min Li**[2]

[1]*School of Physical Education (Main Campus), Zhengzhou University, Zhengzhou, Henan 450001, China*
[2]*English and Economic and Trade Department, Jiaozuo Teachers' College, Jiaozuo, Henan 454000, China*

Correspondence should be addressed to Zhesen Chu; czs@zzu.edu.cn

Received 1 December 2020; Revised 3 February 2021; Accepted 5 February 2021; Published 15 February 2021

Academic Editor: Wei Wang

In this paper, we study the estimation of motion direction prediction for fast motion and propose a threshold-based human target detection algorithm using motion vectors and other data as human target feature information. The motion vectors are partitioned into regions by normalization to form a motion vector field, which is then preprocessed, and then the human body target is detected through its motion vector region block-temporal correlation to detect the human body motion target. The experimental results show that the algorithm is effective in detecting human motion targets in videos with the camera relatively stationary. The algorithm predicts the human body position in the reference frame of the current frame in the video by forward mapping the motion vector of the current frame, then uses the motion vector direction angle histogram as a matching feature, and combines it with a region matching strategy to track the human body target in the predicted region, thus realizing the human body target tracking effect. The algorithm is experimentally proven to effectively track human motion targets in videos with relatively static backgrounds. To address the problem of sample diversity and lack of quantity in a multitarget tracking environment, a generative model based on the conditional variational self-encoder conditional generation of adversarial networks is proposed, and the performance of the generative model is verified using pedestrian reidentification and other datasets, and the experimental results show that the method can take advantage of the advantages of both models to improve the quality of the generated results.

1. Introduction

In recent years, human behavior understanding, as a key task in intelligent applications such as automated driving, service robots, and advanced surveillance systems, is one of the hotspots of interest for researchers in the field of computer vision [1]. An accurate understanding of human behavior is a key prerequisite for human-computer interaction in robots and other intelligent devices, covering multiple stages from perception to representation and analysis of human behavior [2]. In dynamic real-life scenarios, robots need to accurately perceive their surroundings and quickly process information to analyze and learn human behavior to make the right decisions and complete relevant tasks in various situations [3]. Motion trajectory prediction is one of the tasks in pedestrian behavior analysis [4]. Pedestrian trajectory prediction is usually built on top of other tasks in computer vision orientation, such as pedestrian detection, pedestrian attribute recognition, and semantic segmentation, which refers to predicting how agents move over time in scenarios involving multiple agents [5]. Observations, a priori information about the surrounding environment, and information about pedestrian movement are used to locate and predict the location of the target pedestrian in future frames of the video [6]. Predicting the trajectory of pedestrians is of great practical significance; for example, in the safety-critical task of automated driving, only by correctly deducing the intentions of pedestrians around the vehicle and accurately predicting their future trajectories can the vehicle plan its path, avoid obstacles, and prevent crashes from occurring [7]. At the same time, due to the complexity of human behavior and the diversity of the surrounding environment, accurately predicting the trajectory of pedestrians is also a challenging task [8]. The changes of pedestrian trajectories are determined by a variety of factors such as the goal intention, the direction and location of the surrounding

agents, and the information of the scene, some of which cannot be observed directly and need to be inferred from a large number of noisy clues or need to be modelled and learned based on contextual information [9].

Al-Jarrah et al. proposed the Social-LSTM method, one of the earliest deep learning methods for dealing with the problem of pedestrian trajectory prediction, which uses a recurrent neural network to model the motion of observed pedestrians in a video scene and uses the modelling results as a prediction of the future trajectory of the pedestrians [10]. Where the social pooling layer is used to capture the interactions between pedestrians, however, this approach does not consider the importance of different pedestrians [11]. The method of combining Motion Vector (MV) with its associated coding syntax elements is proposed in the literature [12], which is a good indicator of the motion target in the scene. Since this information is in the video bitstream, one would like to try to use it for moving target tracking [13]. The advantages of compressed domain-based tracking methods are their high efficiency and speed [14]. This is because they can avoid excessive decoding of video and storage and processing of pixel values in the pixel field, and they are usually processed in extremely few data centers [15]. The disadvantages are that they are overly dependent on the coding method for compressed video and may reduce accuracy because it is limited by the low-resolution motion sampling grid: typically, an MV uses 4×4 or larger modules/units [16]. The particles change with the detection results [17]. After this, the detect-and-track strategy has become a popular method for multitarget tracking [18]. The basic idea is to use an offline trained detector to detect the target frame by frame and then select a certain time window to correlate the detected results with the trajectory of the target to be tracked. The detection part of the tracking process is the bottleneck of real-time performance, and applying other methods to replace target detection is also a solution to reduce computation time [19]. Therefore, Pierson proposed a localized large density 3D vision hull reconstruction algorithm to replace the detection algorithm in multiobjective tracking, while using particle filtering that incorporates Tyson's polygon method segmentation for tracking [20].

In this paper, we study human motion targets contained in HEVC video coded streams, based on information such as motion vectors partially decoded as a database. The HEVC compressed encoded video processing with the human motion target as the object provides an in-depth study of detection and tracking techniques. In this paper, the visual attention mechanism is investigated. The bottom-up search process can identify local features of visual stimuli, while the top-down attention process is task-based and guided by the global structure of the scene to detect visual saliency. Combining these two global prior and local features can effectively achieve saliency target search. At the same time, it is found that the guided search theory is an attentional model that simulates both top-down global search and bottom-up local features. Therefore, based on the guided search theory, we propose a video salient target detection model based on a two-pathway framework. The model uses spatiotemporal contrast to guide the search for salient targets. First, interframe mappings of colon contrast and motion contrast along a nonselective path, combined with saliency cues from the previous frame, are used as a priori information about the possible spatial location of the target. At the same time, low-level features such as luminance, color, and motion are extracted in the selective pathway to achieve an accurate search of the target. Finally, an improved Bayesian inference model is used to further obtain optimal results. Our algorithm does not require parameters and can automatically detect significant targets in the video.

2. Fast Motion Estimation Model Analysis Design

2.1. Theoretical Model of Feature Integration of Attention. Feature integration theory suggests that features are automatically recorded early on and parallel across the visual field, whereas targets are not individually identified until later, a step that requires focused attention. It is assumed that visual scenes are initially encoded according to several separable dimensions, such as color, orientation, spatial frequency, luminance, and direction of motion [21]. To recombine these separate representations and to ensure the correct synthesis of features for each target in a complex scene, separate stimulus locations are processed consecutively, and attention is focused together. Any features present in the same gaze point were combined into a single object. Reisman argues that, without focused attention, it is impossible to relate features to each other. It seems impossible to consciously "perceive" an individual shape without giving it a color, size, brightness, and position. However, an unattended area is not perceived as space, as shown in Figure 1.

Reisman hypothesizes that the early stages of vision can only detect independent features, including color, size, orientation, contrast, tilt, curvature, and line endpoints, and may also include motion and distance differences. These features are free-floating, not bound to the object to which they belong, and their location is subjectively indeterminate. The perceptual system independently encodes the features of each dimension to form a feature map. The second stage is the focused attention stage, which focuses on details analysis and processing and is the process of integrating various features into a target. The perceptual system correctly associates separate features (color, orientation, size, distance, etc.) to form specific representations of an object. At this stage, it is necessary to locate the features, i.e., to determine where the boundaries of the features are located and form a location map. Processing the location information of a feature requires focused attention [22]. Concentrated attention is like glue, which integrates original, separated features into a single object. This series of processes is somewhat slower than the former. Because more effort is required, when attention is overloaded or people are distracted, especially when the attentional demands are high, the features of the stimulus can be combined inappropriately, resulting in illusory phenomena. Feature integration occurs in the later stages of visual processing and is a nonautomated, sequential process, based on the digital offset

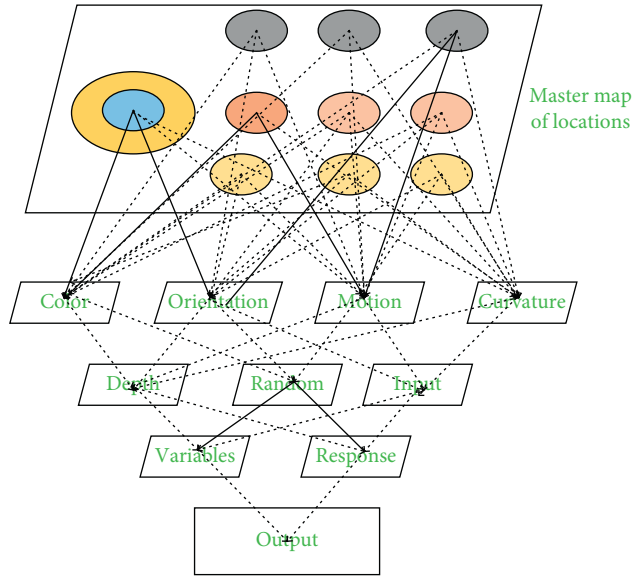

Figure 1: Theoretical model of feature integration.

of the image to compensate for the movement of the digital camera during the shooting of the video, thereby reducing the global motion blur and stabilizing the image in the video.

To avoid changes in brightness due to shadowing effects, the H-S algorithm first assumes that the surface to be imaged is flat. It further assumes that the incident illumination is uniform over the entire surface. Then, the brightness of a point in the image is proportional to the surface reflectance of the corresponding point on the object. Also, it is assumed that the reflectance varies smoothly and that there are no spatial discontinuities. The H-S algorithm also excludes the case where objects block each other, partly because discontinuities in reflectance are found at the object boundaries. Thus, in the simple case of these assumptions, the motion of the luminance pattern in the image is directly determined by the motion of the corresponding points on the surface of the object. Once the flow of light is known, calculating the velocity of the points on the object is a simple geometric problem.

If each point of the image brightness pattern is moving independently, there is little hope of recovering the visual speed. More often, we see opaque objects of finite size subjected to rigid motion or deformation [23]. In this case, the neighboring points on the object have similar velocities, and the velocity field of the luminance mode changes smoothly almost everywhere in the image. In the case where one object blocks another, a discontinuity in light flow may occur. Therefore, Horn and Schenck argued that one way to express additional constraints is to minimize the square of the size of the velocity gradient of the optical flow. The H-S algorithm requires the optical flow to be as smooth as possible, so its constraints are minimized, and equation (1) is the smooth constraint equation for the H-S algorithm.

$$ {}_{a}^{G}H_{t}^{v}S(t) = \lim_{h\longrightarrow 0} \frac{1}{h^{v}} \sum_{m=0}^{[((t-a)/h)]} (-1)^{m} \frac{\Gamma^{2}(v)}{m!\Gamma(v-m)} f\left(t-m^{h}\right). \tag{1} $$

According to the fundamental equation of the optical flow vector, the optical flow error is minimized to

$$ \mathrm{STFT}[A(t,w)] = \int_{0}^{+\infty} a^{2}(\tau) g^{2}(\tau - t) e^{-jw\tau} d^{2}\tau. \tag{2} $$

Thus, the solution to the optical flow field is transformed into a solution to the following equation:

$$ M^{*} = \arg\max \left\{ \sup_{x\in\Omega_{x}} \left| b(x) - W^{T} S\left(x^{3}\right) \right| \right\}. \tag{3} $$

The normalization of motion vectors caused by a different reference frame to a neighboring reference frame is called time-domain normalization. Also, since the coding units of HEVC codes are not equally divided, therefore, it is necessary to plan their division sizes uniformly, and here they are all divided by the smallest coding unit, which is the 4×4 standard [24]. In addition to time-domain normalization, spatial normalization, which is the process of dividing coding units of different sizes into equally sized blocks and assigning them motion vectors, is also required. The following is the formula for the motion vector median filter processing in this chapter, where M is the set of all

motion vectors in the frame and Q is the set of motion vectors after X (Vector Median Filter) processing.

$$Q_N(w) = \frac{Q_N(w)}{Q_L(w)} = \frac{t^r r\left(w^T XLX^{Tw} + w^T X_l X_l^T X^{Tw}\right)}{t^r r\left(w^T XL_N X_w^{Tw} + w^T X_b^a X_b^T X^{Tw}\right)}. \tag{4}$$

However, due to some degree of shaking in the background of some videos, the global motion vector has a large error, and the human motion target region cannot be detected directly, so global motion estimation of video frames is needed to solve this problem. The problem of motion vector enhancement due to shaking is mainly predicted by global motion estimation, which can obtain a more reliable vector field of the foreground motion target. The divergence of the curl of the vector field is always 0; that is, the curl field of the vector field must be a passive field.

$$\phi_{m,n} = \frac{\left\|q_m^n\right\|^2}{\delta^2} \exp\left(-\frac{\left(q_m^n * z\right)^2}{2\delta^2}\right) * \left[e^{i\left(q_{m,n} * z\right)} - e^{-\delta^2/2}\right]. \tag{5}$$

After the global motion compensation for each pixel frame in the video, the motion vector field needs to be accumulated to improve the accuracy and reliability of moving target detection. The main purpose of motion vector field accumulation is to eliminate background-induced noise. However, in an intelligent surveillance video, human motion targets may be close to each other and cannot be detected by motion vector field accumulation alone. Therefore, it is necessary to add a backward iterative projection accumulation method. Equation (6) is a mathematical expression for backward projection.

$$\frac{\partial L}{\partial \beta_j} = \sum_{i=1}^{n}\left[y_i - \frac{\exp\left(a + \sum_{j=1}^{m} x_{ij}^n \beta_j\right)}{1 + \exp\left(a + \sum_{j=1}^{m} x_{ij}^m \beta_j\right)} x_{ij}\right] = 0, \quad j = 1, 2, \ldots, m. \tag{6}$$

Threshold processing techniques are widely used for the segmentation and detection of motion video. In compressed domain video, the threshold-based target detection method mainly consists of motion vector blocks. In general, in video sequences where the camera is relatively stationary, the background vector amplitude is small or zero. However, in video sequences where the camera is moving, the background vector usually has a certain amplitude, which will be studied in this chapter for video sequences where the camera is relatively stationary.

$$g_i = \frac{\omega_i^2}{\sum_{j=1}^{n} \omega_i^2}, \sum_{i=1}^{n} g_i = 0, \tag{7}$$

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}} = \frac{\lambda_{\min} - n}{n^3 - 1} \times \frac{1}{R^T I}, \tag{8}$$

$$P\left(y_i\right) = p_i^{y_i}\left(1 - p_i\right)^{1 - y_i^i}. \tag{9}$$

In the online multitarget tracking process, to obtain the continuous trajectory of a particular target, it is necessary to find the exact location of all the targets to be tracked in that image, while the tracker is acquiring the next frame of image information, and the process of finding the location of a particular target is called bounding box regression. These four mappings can usually be obtained by linear regression. For the multiobject tracking task, this paper firstly extracts the depth features of the whole image through the convolutional neural network, inputs the acquired depth features into the linear regression algorithm, and then outputs the above four mappings after the calculation of the linear regression algorithm.

2.2. Fast Motion Estimation Design. The CNN-LSTM pedestrian trajectory prediction model proposed in this paper is shown in Figure 2, and the network framework is a codec structure with a one-dimensional convolutional-inverse convolutional network for both encoder and decoder. As a common network pattern in deep learning, the convolutional-inverse convolutional network is often used to handle image segmentation, generation, classification, and other image domain tasks. The process of learning data from a neural network is fundamentally a linear transformation operation that multiplies the weight matrix of the network with the input multidimensional vector to obtain another multidimensional vector. There are too many learning algorithms, such as classification, regression, clustering, recommendation, and image recognition. It is not easy to find a suitable algorithm, so, in practical applications, we generally use heuristic learning methods to experiment. Usually, at the beginning, we will choose algorithms that everyone generally agrees on, such as SVM, GBDT, and Adobos. Now, deep learning is extremely popular, and neural networks are also a good choice. If you care about accuracy, the best way is to test each algorithm one by one through cross-validation, compare them, then adjust the parameters to ensure that each algorithm reaches the optimal solution, and finally choose the best one. Convolutional networks are mainly used to extract low-dimensional features from high-dimensional vectors, which is equivalent to an encoder; inverse convolutional networks are the opposite, reconstructing high-dimensional vectors from low-dimensional features, which is equivalent to a decoder, i.e., performing the forward and backward propagation operations of the convolutional neural network in reverse. In this paper, in the codec module of the model, the historical information on the position and size of pedestrians observed in the first-view video and the self-motion information of the camera is encoded by constructing a convolutional-inverse convolutional network, from which the features of the pedestrian position and size information and the self-motion information of the camera are extracted, and the

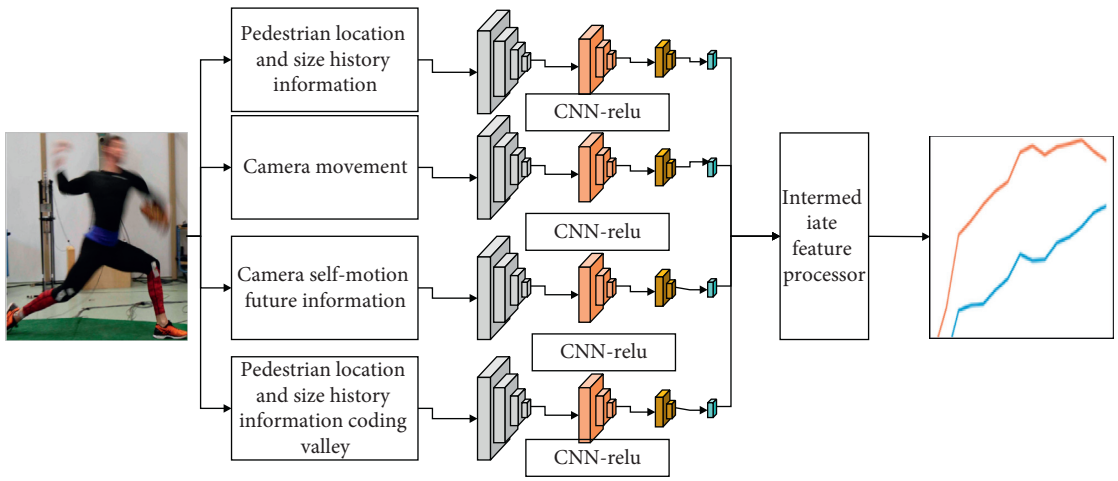

Figure 2: CNN-LSTM trajectory prediction model.

input and output streams are in the form of one-dimensional convolution.

A sequence of pedestrian location and size history information is of size $4 * 10$. The output of the first one-dimensional convolution layer is input to the second one-dimensional convolution layer, the output of the second one-dimensional convolution layer is input to the third one-dimensional convolution layer, and finally, the output of the third one-dimensional convolution layer is input to the fourth one-dimensional convolution layer. In this encoder, the output results of each layer are BN batch normalized and activated by the ReLU activation function. The network architecture configuration of the motion position and size history encoder is shown in Table 1.

To evaluate the accuracy of the model, the final displacement error (FDE) and the average final displacement error FDE are the distance between the destination of the predicted trajectory and the destination of the real trajectory. Also, since this paper predicts the location and size of pedestrian detection frames, U is introduced as an evaluation index, and MJ is a common index used in target detection tasks, which refers to the overlap ratio between the detection frames predicted by the model and the original marked detection frames, i.e., the intersection of the detection results and the true values. The ratio of the union between the two is used in this paper to evaluate the predictive performance of the location and size of the future detection frame for pedestrians.

The changes in the loss values during the training of the model on the three data sets are shown in Figure 3. The horizontal axis is the training period, and the vertical axis is the log of the loss value. According to these three figures, when the model is first trained because the model has not learned the sample data, the difference between the generated prediction sample and the real sample is large, resulting in a large loss value, which drops sharply when the training period is less than 100 and slows down after 100. In each dataset, for each cross-validation model training, the overall trend is that the model loss function decreases and converges as the number of training sessions increases, and the model tends to stabilize.

During the training process, the neural network must backpropagate the residuals, and during this backpropagation process, the phenomenon of gradient explosion and gradient instability with the gradient disappearing often occurs. The gradient explosion problem occurs when the learning rate of the hidden layer in front of the neural network is higher than that of the hidden layer behind, resulting in faster network changes; the gradient disappearance problem occurs when the learning rate of the hidden layer in front of the neural network is lower than that of the hidden layer behind, resulting in slower network changes. The root cause is that the neural network uses chain derivation to solve the gradient of each layer, and the multiplication process in the middle may make the residual calculation unstable, which may prevent the model from further training.

The network is fine-tuned for extracting deep appearance features of pedestrians by constructing a residual network fused with the SE module, comparing different loss functions, learning the metric of different pedestrian targets through the twin network framework, and learning the appearance feature representation. The network is then fine-tuned to extract the deep appearance of pedestrians and learn the appearance representation. The method framework is shown in Figure 4.

Briefly, guided search theory suggests that, in the early stages of the visual system, all locations are processed in parallel, but only a limited amount of information can be extracted from the visual input. Subsequent processes can perform other, more complex tasks, but only one location or a few spatial locations at a time. The information gathered by the parallel front end is used to reconstruct the deployment of those parts of the visual field that are most likely to contain objects of interest. This guidance is not perfect, but it is much more effective than a random distraction. The

Table 1: Motion position and size history encoder network architecture configuration table.

Network layer type	Number of channels	Convolution kernel size	Output size
Enter	—	0.15	1.2
ID convolution + BN + ReLU	4	0.1	0.4
ID convolution + BN + ReLU	2	0.45	0.57
ID convolution + BN + ReLU	5	1.2	0.58
ID convolution + BN + ReLU	6	2.1	0.89

nonselective pathway is a form of late selection, in which the process moves to an advanced state before any "bottlenecks" are encountered.

The selective pathway embodies early selection, with little processing before the bottleneck is reached. Traditionally, these were competing alternatives, and here, they coexist. However, traditional delayed selection allows for target recognition (e.g., word recognition) before the bottleneck. Nonselective paths, while capable of extracting some semantic information from the scene, do not have recognition capabilities. Based on the fundamentals of guided search theory, we propose a model for video salient target detection using comparative information in the temporal and spatial domains as an a priori guided visual search, considering the strong correlation between video frames in our analysis. Among them, the local features extracted by the local feature extraction algorithm (such as SIFT) are called descriptors. For example, the dimension of the SIFT descriptor is 128, and then if m descriptors are extracted from an image, the descriptor matrix of the image is $m * 128$. The number of local features extracted from each picture may be different, so we need to merge these different numbers of descriptors into a feature vector (assuming the dimension is n) to represent the entire image, so that a picture can use a $1 * k$ that is represented by a vector. After doing so, image retrieval and classification tasks can be easily realized.

The core idea of introducing a state machine is to use different attitude fitting methods in different motion states to improve the accuracy of attitude fitting. In this paper, a finite state machine is constructed for five motion states: normal walking, fast walking, running, feet standing still, and jumping, and different posture-fitting methods are implemented in different states. The posture changes in normal walking are regular, and feet touch the ground with obvious periodicity. In fast walking, the muscles of the legs cause a large error to the sensor, and the jittering of the muscles when landing on the ground causes the sensor to jitter, and the foot touchdown time is short. Running is a more complex sport, and its most obvious characteristic is the extremely short foot touchdown time. The conditions for switching between states need to find out the threshold or feature conditions for the

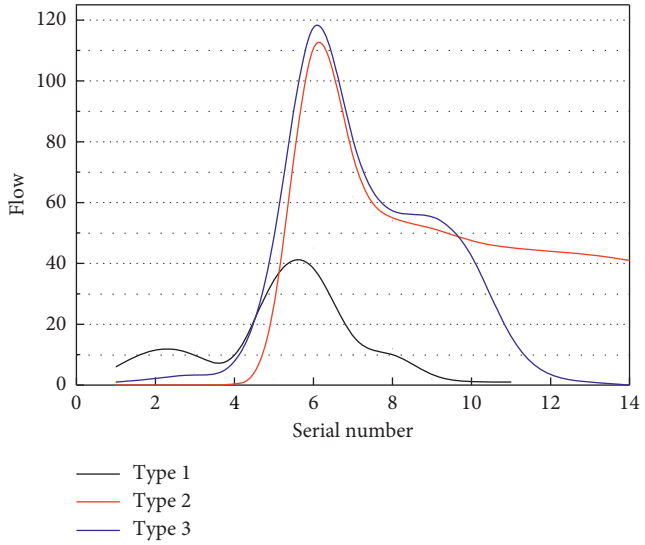

Figure 3: Training loss function diagram of CNN-LSTM model in MOT16 dataset.

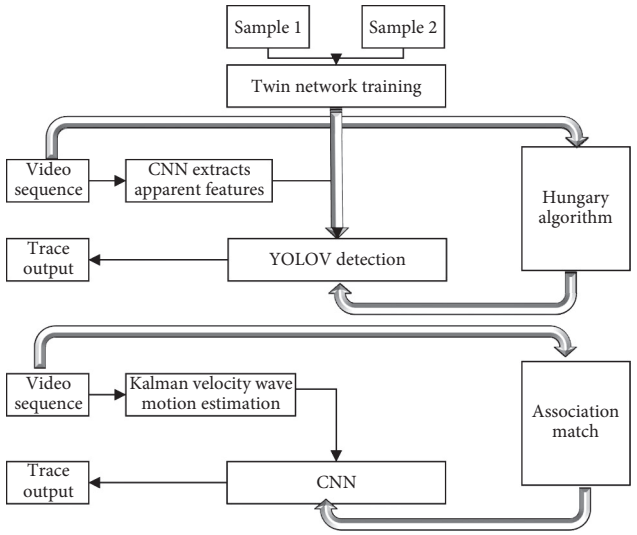

Figure 4: Framework diagram of a motion tracking method incorporating apparent features.

above actions and analyze the angular velocity characteristics, acceleration characteristics, and joint angles in different movement state to determine which posture fitting method to use at a certain moment.

3. Results and Analysis

3.1. Comparative Experimental Analysis. Figure 5 shows the HEVC standard test video sequence of Basketball Drill and Bimal, both of which have relatively static backgrounds, 382×480 resolution, multiple human motion targets, and large human motion targets in the video. The GMM and VIBE algorithms are used for comparison with the algorithm in this paper, in which both algorithms are pixel-based target detection algorithms, and to avoid the visual interference caused by color, grayscale images are used for comparison.

Since the GMM hybrid Gaussian model algorithm is susceptible to interference from light intensity, there is a lot of noise when detecting human motion targets in the images, and the detection rate of human targets is relatively poor due to the large difference between the human body and the background. The VIBE algorithm is relatively poor at detecting the integrity of the human motion target. Since this paper adopts motion vector field accumulation and thresholding of its time-domain correlation, there is no noise in the background part, and the detection performance is good. The algorithm proposed in this chapter is compared and analyzed by three test criteria: accuracy (hereinafter referred to as *P*-value), recall (hereinafter referred to as R-value), and F-measure (hereinafter referred to as F-value).

Figure 6 shows a comparison of the three test criteria for the Basketball Drill video sequence. The first 100 frames of the three sequences are selected for statistics, and the detection results of GMM and VIBE are relatively poor, while the detection results of the method in this paper are relatively good. The above analysis shows that this method significantly reduces the detection time with less precision loss, has relatively good real-time performance, and is better than the GMM and VIBE algorithms in human target detection.

The proposed method has higher R-values, as well as F-values, especially R-values, than the other two methods, but it slightly lacks detection accuracy because it is based on the detection of human targets in the compressed domain, which has a larger basic unit than the pixel domain, thus resulting in a lower *P*-value. This comparative analysis shows that the method greatly reduces the detection time, and although the accuracy is reduced, the real-time performance is better. A threshold-based human body target detection algorithm is proposed, which consists of several steps, by normalizing the initial motion vector. The motion vector field accumulation is performed, and finally, the human motion target part of the image is detected in combination with the set threshold value. After the above steps, the video sequences are finally selected for experiments, and the experimental results show that the algorithm in this chapter is effective in detecting human body targets. The advantages are that the algorithm is simple, real-time, and fast. However, the requirements for the video sequence are more stringent, requiring a relatively static background, and the detection effect depends on the selection of the threshold value.

People are not interested in all the information in the whole image, but only in some regions or parts of the image, which are the Regions of Interest (ROI). If we can identify these regions and assign different priorities to different regions for processing, the efficiency and accuracy of image processing will be greatly improved. The underlying features of the image directly affect the quality of ROI extraction; therefore, it is necessary to analyze and study the degree of influence of each underlying feature on the ROI of the extracted image. Since the CGVS algorithm is a significant target detection model for images, we divide the video sequence into frames and then use the CGVS algorithm to compute the significance map for each frame.

3.2. Analysis of Trend Prediction Results for Direction of Motion. For smaller targets in the field of view, due to the absence of detailed appearance features, the error in extracting their body features is large, so the motion trend is only determined by the movement pattern of the pedestrian frame in the previous frame; i.e., the motion trend is only predicted by the velocity model. In the current frame, the motion speed of the pedestrian frame (i.e., the number of frames between which the displacement/interval of the target frame is to be predicted) is decomposed in two directions horizontally and vertically to the field of view, and the horizontal and vertical velocities are input into the threshold loop unit for learning. The training is performed intensively, and the training data is the extracted motion state of each pedestrian target in both horizontal and vertical directions. The trained model predicts the trend of the target's movement in both horizontal and vertical axes. The model predictions are classified and output through a simple SoftMax layer, and the velocity predictions in both axes are integrated to estimate the overall displacement of the frame. There are two models of this approach, a larger target motion trend prediction model based on attitude estimation and a smaller target motion trend prediction based on velocity estimation. Figure 7 shows the overall results of the ablation experiment and the performance validation results of the tracking algorithm after integrating the motion estimation model.

For the pose estimation part, the network structure uses a threshold cyclic unit with 1024 implicit nodes and does not use a representation learning layer independent of the time series information for the appearance feature processing. 1024 nodes of the training unit have a significant impact on the model speed and exponentially increase the difficulty of training the model as the sequence length increases. Therefore, we use an encoder network to reduce the training node dimension to 54 (corresponding to the number of individual human joints provided by the pretrained model on the adopted dataset). In the testing process, for larger targets, the contours of the target are first extracted using a fast instance segmentation method, and then the pedestrian pose is obtained by clustering and finally input to a based

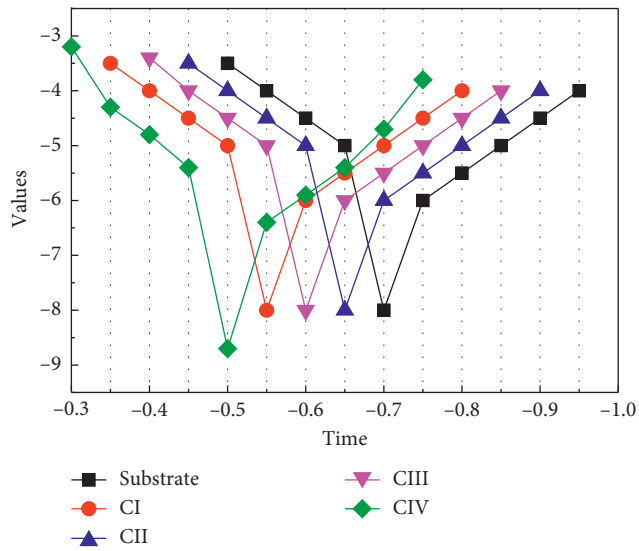

Figure 5: Diagram of the main errors in tracking.

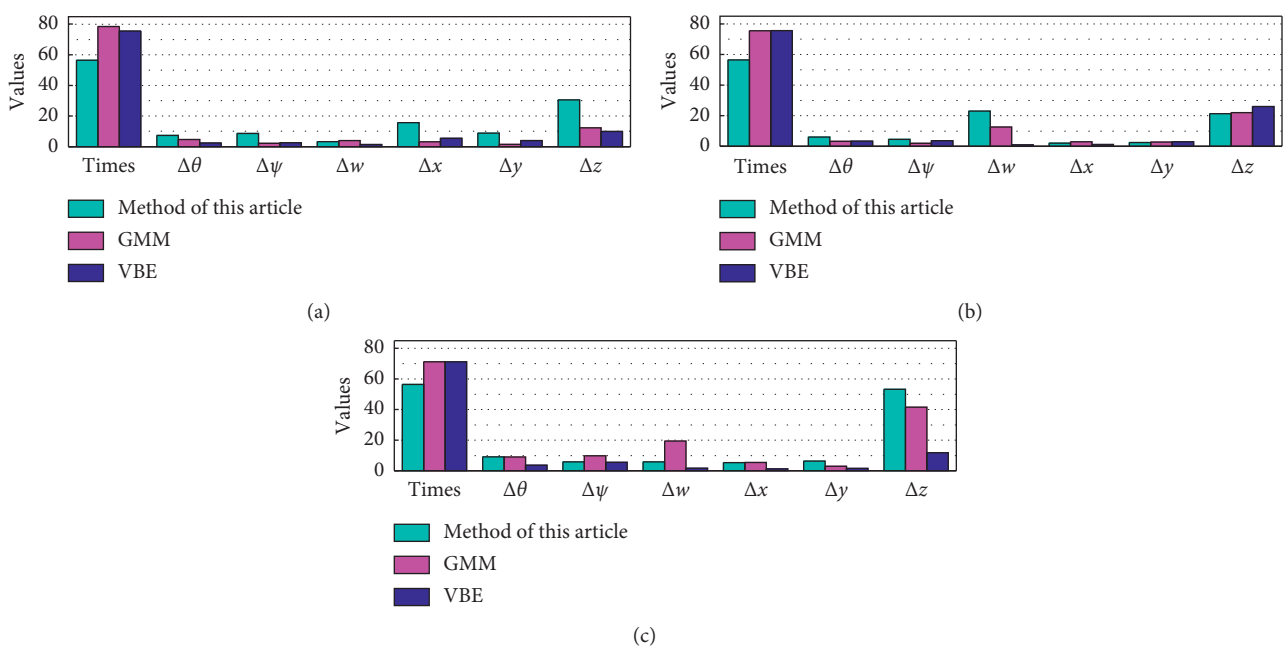

Figure 6: Curve comparison chart.

pose prediction model for motion trend prediction. For the velocity estimation part, the network structure consists of a threshold cyclic unit with 300 implicit nodes. Training is performed based on the pedestrian target motion velocity extracted from the MOT Challenge public dataset, optimized by the RMSprop method with momentum.

The GRU-based velocity estimation model is evaluated since, in the application of the overall tracking algorithm, the GRU-based velocity estimation model is only used for small targets in the scene. To validate the method during the ablation experiments, the GRU-based velocity estimation model is extended to all targets in the field of view. The experimental results are shown in Figure 8, and the performance is evaluated using the same metrics as in the previous subsection. It can be seen from the table that the GRU-based velocity estimation model, although broader than the based attitude estimation model for the target case, is less effective overall, showing significant improvement only in the IDF metric, while the MOTA and MT metrics are almost unchanged. For the ML index, the number of missing targets tends to increase slightly. The reason for this is that the GRU-based velocity estimation model does not use the appearance feature as a supervisor, but only learns the border position of the sequential frame of the target, and

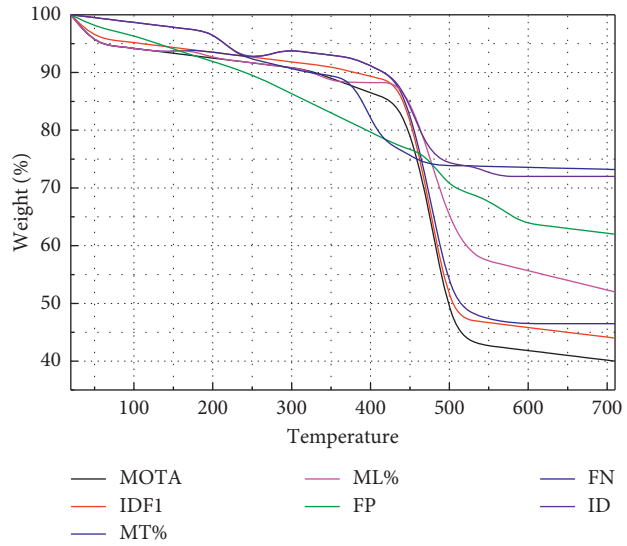

Figure 7: Motion estimation model for single-objective tracking-based augmented multiobjective tracking method and boundary-box regression.

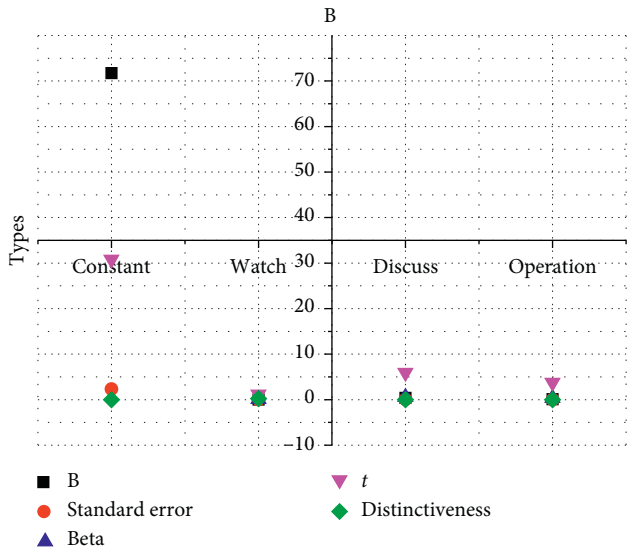

Figure 8: Attitude estimation model performance improvement.

then predicts the position after missing multiple frames. Targets in the field of view are more likely to have similar border sizes and the same motion trend without considering the appearance features. In this case, the velocity estimation model is unable to determine whether the targets appearing after multiple frames are the original lost targets or not and then correlate them uniformly, which causes new targets that should have been initialized to be recognized as lost targets, resulting in more lost targets and more false detections. If the basic unit is larger, the P value will become smaller and smaller, so that the result becomes more and more accurate.

The reason for the poor overall effect is that the quality of the initial data is not extremely good. Under nonstrenuous

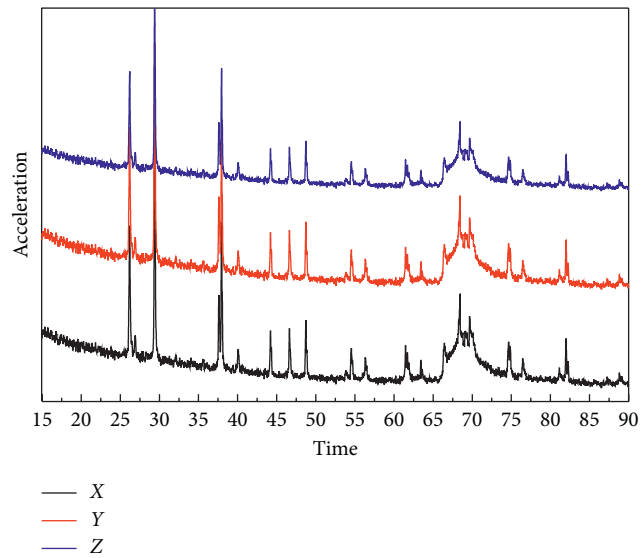

Figure 9: Changes in left foot acceleration and angular velocity during jumping.

jumping, the height signal of the crotch changes more but less frequently, and the acceleration and angular velocity of the foot change drastically at the instant of a touchdown but with a longer period of change. In the jumping state, the acceleration and angular velocity signals from the left foot inertial sensor are collected, as shown in Figure 9.

The difference in acceleration peaks between the running and jumping states is small, but the difference in frequency is large; i.e., the difference in wavelength is large. The wavelength of the signal is determined by calculating the distance between the two peaks to classify the jumping state. Fit the collected discrete data points to a curve, solve the analytical expression of the curve by numerical fitting, find the extreme points by finding the derivative function of the expression, and then determine the peaks or troughs. The time difference between the two is calculated and compared to a threshold value. If the time difference is greater than the threshold value, it indicates a jumping state. First, the carrier, navigational, and world coordinate systems, as well as the transformations between them, are introduced, and then different posture representation methods are analyzed, mainly the quadratic number method, the directional cosine matrix method, and the Euler angle method, as well as the feature comparisons and transformations between them. Afterwards, based on the analysis of the human lower limb motion model and model errors, several ways of posture fitting are focused on, including posture fitting based on forwarding kinematics, posture fitting based on inverse kinematics, posture fitting based on integral displacement, and complimentary fitting based on Kalman filtering algorithm with forward and inverse kinematics. Finally, a finite state machine- (FSM-) based posture fitting method is used to switch between different postures in real-time under different motion states, and the effectiveness and feasibility of the posture fitting method are verified experimentally.

4. Conclusion

In this paper, a feature matching tracking method based on a prediction mechanism is proposed to address the multitarget human tracking problem. First, the human body locations and areas are matched according to the proposed threshold-based detection algorithm. Then, the area overlap ratio is used to determine whether the tracking match succeeds or fails. The human motion target that fails to be matched is tracked again by human matching. To solve the problem of human target tracking failure due to occlusion or missed detection, this paper combines motion vectors with Kalman filtering and prediction mechanism to predict and match the human position information due to missed detection and performs human tracking counting according to the human matching state in the area. We combine the significant map of the previous frame a priori of the current frame to guide the search of the selective pathway. In the selective pathway, we compute the luminance, color, motion amplitude, and motion direction of the underlying features to accurately estimate the salient targets. Finally, a Bayesian model is used to integrate the global prior information with the local underlying features, and in the Bayesian model, we further add the geophone distance, which can effectively suppress the background. Finally, we compare our algorithm on three widely used video salient target detection datasets and demonstrate that our algorithm can achieve the video salient target detection task completely and accurately and is robust to a variety of more complex scenarios. By integrating this method into the multiobjective tracking algorithm proposed in this paper, the target identification accuracy metrics that characterize long-time tracking performance and the ability to handle occlusion are improved by 2.5% and 3%, respectively, and the target identification exchange metrics are reduced by 21.4% and 15.9%, respectively, with 229% and 400% increase in operating speed, respectively, without being affected.

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 known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Jiang, L. Yan, Y. Xia, Q. Guo, M. Fu, and K. Lu, "Asynchronous multirate multisensor data fusion over unreliable measurements with correlated noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2427–2437, 2017.

[2] H. Wu, Z. Zhang, C. Jiao, C. Li, and T. Q. S. Quek, "Learn to sense: a meta-learning-based sensing and fusion framework for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8215–8227, 2019.

[3] P. Ghamisi, R. Gloaguen, P. M. Atkinson et al., "Multisource and multitemporal data fusion in remote sensing: a comprehensive review of the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 1, pp. 6–39, 2019.

[4] H. Zhang, X. Zhou, and Z. Wang, "Adaptive consensus-based distributed target tracking with dynamic cluster in sensor networks," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1580–1591, 2018.

[5] X. Yuan and Y. Pu, "Parallel lensless compressive imaging via deep convolutional neural networks," *Optics Express*, vol. 26, no. 2, pp. 1962–1977, 2018.

[6] D. Nada, M. Bousbia-Salah, and M. Bettayeb, "Multi-sensor data fusion for wheelchair position estimation with unscented Kalman Filter," *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 207–217, 2018.

[7] V. Radu, C. Tong, S. Bhattacharya et al., "Multimodal deep learning for activity and context recognition," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–27, 2018.

[8] Q. Zhou and Y. Zheng, "Long link wireless sensor routing optimization based on improved adaptive ant colony algorithm," *International Journal of Wireless Information Networks*, vol. 27, no. 2, pp. 241–252, 2020.

[9] Z. Zhao, X. Wang, and T. Wang, "A novel measurement data classification algorithm based on SVM for tracking closely spaced targets," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 4, pp. 1089–1100, 2018.

[10] M. A. Al-Jarrah, A. Al-Dweik, and M. Kalil, "Decision fusion in distributed cooperative wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 797–811, 2018.

[11] M. Pourshamsi, M. Garcia, M. Lavalle, and H. Balzter, "A machine-learning approach to PolInSAR and LiDAR data fusion for improved tropical forest canopy height estimation using NASA AfriSAR Campaign data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 10, pp. 3453–3463, 2018.

[12] N. Mittal, U. Singh, R. Salgotra, and B. S. Sohi, "An energy efficient stable clustering approach using fuzzy extended grey wolf optimization algorithm for WSNs," *Wireless Networks*, vol. 25, no. 8, pp. 5151–5172, 2019.

[13] W. Huang, Y. Ling, and W. Zhou, "An improved LEACH routing algorithm for wireless sensor network," *International Journal of Wireless Information Networks*, vol. 25, no. 3, pp. 323–331, 2018.

[14] S. Nakayama, G. Blacquière, and T. Ishiyama, "Automated survey design for blended acquisition with irregular spatial sampling via the integration of a metaheuristic and deep learning," *Geophysics*, vol. 84, no. 4, pp. P47–P60, 2019.

[15] J. Hülsmann, J. Traub, and V. Markl, "Demand-based sensor data gathering with multi-query optimization," in *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2801–2804, 2020.

[16] A. Belhadi, Y. Djenouri, J. C.-W. Lin, and A. Cano, "Trajectory outlier detection," *ACM Transactions on Management Information Systems*, vol. 11, no. 3, pp. 1–29, 2020.

[17] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: model-based, AI-based, or both?," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.

[18] A. B. Hamida, A. Benoit, and P. Lambert, "3-D deep learning approach for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

[19] A. Farasat, G. Gross, R. Nagi, and A. G. Nikolaev, "Social network analysis with data fusion," *IEEE Transactions on Computational Social Systems*, vol. 3, no. 2, pp. 88–99, 2016.

[20] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: a review of recent research," *Advanced Robotics*, vol. 31, no. 16, pp. 821–835, 2017.

[21] L. Li, K. Ota, and M. Dong, "Deep learning for smart industry: efficient manufacture inspection system with fog computing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4665–4673, 2018.

[22] A. Jalali and H. Farsi, "A new steganography algorithm based on video sparse representation," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 1821–1846, 2020.

[23] H. Song, J. J. Thiagarajan, P. Sattigeri, and A. Spanias, "Optimizing kernel machines using deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5528–5540, 2018.

[24] L. Bruzzone and A. Bhattacharya, "A novel technique based on deep learning and a synthetic target database for classification of urban areas in PolSAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 1, pp. 154–170, 2017.