


## RESEARCH ARTICLE

# Design of Human-Computer Interaction Gesture Tracking Model Based on Improved PSO and KCF Algorithms

DINGHUA HE<sup>1</sup>, YAN YANG<sup>1</sup>, AND RANGZHONG WU<sup>1,2</sup> <sup>1</sup>School of Information Technology Application and Innovation, Wuhan Polytechnic, Wuhan 430074, China<sup>2</sup>School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China

Corresponding author: Rangzhong Wu (wurangzhong1991@163.com)


**ABSTRACT** The detection and tracking of gesture targets is an important aspect in the dynamic gesture recognition. To meet the accuracy and speed requirements of human-computer interaction for dynamic gesture recognition, this study explores long-term gesture recognition under monocular RGB cameras. This study uses an improved particle swarm optimization algorithm as the feature extraction method, and introduces a mixed Gaussian model and kernel correlation filtering to complete gesture detection and tracking. And it has constructed a dynamic gesture tracking model on the ground of kernel correlation filtering. The experimental results show that the skin color based gesture detection algorithm has the minimum average relative error value of 0.321 on different datasets, with accuracy and recall rates higher than 0.8. The maximum correlation coefficient R-squared value is 0.823, and the detection speed reaches 36.32 frames per second. And this detection method has high repeatability on different datasets, with better detection accuracy for different gesture targets. The F1 value of the gesture tracking model has the largest area of the receiver operation characteristic curve, and the two error values of the model are small, resulting in better gesture tracking performance. In human-computer interaction systems, the detection accuracy and target rejection rate of this method have been significantly improved, and the subjective evaluation of the interaction system by the subjects is relatively high, resulting in good application effects. This study enriches the theoretical foundation of dynamic gesture detection and tracking technology, and improves the quality level of gesture tracking in the field of human-computer interaction. This helps to expand the application scope of human-computer interaction.

**INDEX TERMS** Human-computer interaction, particle swarm optimization, skin tone model, kernel correlation filtering, dynamic gesture detection, hand tracking.

## I. INTRODUCTION

Human-computer interaction (HCI) is the process of exchanging information and instructions between humans and computer systems, allowing computer systems to understand and respond to human needs and instructions through various interaction methods such as posture, voice, and touch [1]. As the boost of artificial intelligence and machine learning technology, the technology of HCI has been continuously optimized and innovated. It is widely

used in fields such as smart mobile devices, augmented and virtual reality, smart homes and the Internet of Things, and healthcare [2]. Gesture recognition (GR) technology plays an important role in HCI, where users control devices through hand movements and postures, making it an intuitive and natural way of interaction. GR improves the user's immersive experience and provides a more intuitive and convenient way of operation for HCI [3]. GR technology can be divided into static GR and dynamic GR. Static GR is the processing and recognition of a single gesture image. Dynamic GR is the recognition of gesture groups within a time period, which requires processing image sequences and

The associate editor coordinating the review of this manuscript and approving it for publication was Giuseppe Desolda .

more complex GR [4], [5]. At present, technologies such as neural networks, support vector machines, nearest neighbors, and distributed local linear embedding have been used to handle various GR. However, in the face of complex and ever-changing human gestures, existing GR technologies still need to strengthen their universality and robustness [6], [7]. The gesture actions of users are often affected by posture changes and occlusion, which challenges the real-time and accuracy of interactive systems in completing user gestures in different complex scenarios [8]. For enhancing the efficiency of dynamic GR, research was conducted on the detection and tracking of long-term changing gestures under monocular RGB cameras. Firstly, linear inertia weights are introduced in the Particle Swarm Optimization (PSO) algorithm to optimize video image features. Then, on the ground of the mixed Gaussian model for skin color detection, gesture detection of RGB images was completed. On this basis, the improved confidence model's Kernelized Correlation Filter (KCF) was used to track dynamic gestures. The research enriches the theoretical research and application technologies in the field of dynamic GR and HCI. The research consists of four parts. The first part is an overview of the current research status of GR and tracking related technologies. The second part proposes a dynamic gesture detection and tracking model for research and design. The third part conducted testing experiments and analysis on the performance of gesture detection and tracking algorithms. The fourth part summarizes and summarizes the experimental results. This study is expected to improve the technical weaknesses of dynamic GR and promote the development of HCI.

## II. RELATED WORKS

GR and tracking are important research directions in the fields of computer vision and HCI, and have received widespread attention worldwide. At present, great progress has been achieved in GR and tracking technology, and researchers have proposed many effective methods to solve the problem of GR. The local processing function of wearable devices on the ground of surface electromyography monitoring muscle activity cannot achieve real-time training and updating of GR models, weakening the effectiveness of GR. Moin et al. designed a wearable surface electromyography biosensing system with sensor based adaptive learning function and a neural inspired hyper dimensional computing algorithm for GR. This algorithm updates the model under different arm positions and sensor replacement conditions. After verification by participants, the system still has a recognition accuracy of 92.87 for GR in 21 cases [9]. Deep learning techniques on the ground of surface electromyography signals are widely used in gesture classification and recognition. However, due to the limitations of multi class GR datasets, the recognition accuracy of existing methods cannot meet the requirements of multi gesture interaction scenarios. Regarding this, Jiang et al. first collected a multi

class dataset of 20 gestures using wearable devices that can obtain surface electromyography and inertial signals, and then designed an improved dual stream deep learning model for GR. The experiment showcases that the accuracy of the dual stream recognition model combining the Transformer module with convolutional neural network has been improved from 71.86% to 98.96% [10]. To improve the application of gesture detection and tracking technology, Yadav et al. first designed a three-dimensional information algorithm for detecting gesture objects by combining color, motion, and AlexNet, and implemented gesture object tracking using AlexNet and point tracker. Then, on the ground of deep convolutional neural networks, gesture trajectory recognition was achieved, and the designed model and algorithm achieved significant improvements compared to the baseline model on different open competition datasets [11]. GR technology is key to driving the application of virtual reality. Wong et al. designed an effective, low-cost capacitive GR sensor. The study uses a median filter as the output of the sensor and selects error correction output code support vector machine and K-nearest neighbor algorithm as the GR classifier. Then it introduces feature compression methods derived from correlation analysis for decreasing the complexity of recognition algorithms. The experimental results show that both machine learning algorithms have high recognition rates [12].

To develop a sustainable, intelligent, and reliable GR system, Ansar et al. implemented gesture localization and automatic labeling in RGB images on the ground of fusion and directional image methods. It utilizes the Grey Wolf algorithm to optimize point features and full hand features in GR, and adopts a reweighted genetic algorithm for gesture classification and recognition. The robustness and effectiveness of this method have been demonstrated through validation on five publicly available datasets [13]. To address the challenges of dynamic GR technology, Gao et al. designed a dynamic GR method using 3D hand pose estimation, data fusion, and deep neural network technology. This method improves the 2D hand pose estimation method on the ground of OpenPose, and achieves the fusion of gesture RGB, depth, and 3D skeleton data using a weighted fusion method. After verification by a dynamic gesture database, the recognition accuracy of this method reaches 92.4% [14]. There are technical difficulties in detecting and tracking the first frame gesture object in a dynamic environment. Yadav et al. designed a region based detection method that utilizes skin color and motion information to detect hand like regions. And using smooth trajectory training, a deep neural network can recognize 60 types of isolated dynamic GR. Verified by a gesture database, the highest accuracy of this method for bare hand detection is 97.80%, which is a relative improvement of 17% compared to the baseline model [15]. GR has an extensive range of utilization in the field of interface based device control, and the demand for touch avoidance interaction technology is constantly increasing. To cope with the complex large-scale

changes in gestures and accurately extract gesture features, Thabet et al. used video sequences from the IBGHG dataset to fuse embedded compact covariance matrix technology into local features on the ground of Gabor Canny Hog features for feature-based gesture tracking. The experiment showcases that the GR accuracy of this method is as high as 96.97% [16]. To address the challenges of GR under lighting changes and background occlusion, Saboo et al. combined color and motion information to complete hand detection. Then, on the ground of the improved sparse optical flow tracking algorithm and Camshift algorithm, gesture tracking was achieved. The experiment showcases that this method has a significant improvement in gesture tracking compared to traditional methods [17]. The use of electromagnetic signals in ultra wideband technology for GR often leads to issues such as radar clutter, signal coupling, and interference. Therefore, Li et al. designed a GR interaction system between humans and smart homes using convolutional neural networks and squeezing and excitation modules. Compared with four different baseline models, this method achieves a gesture activity recognition accuracy of 99.48% and has high anti-interference and robustness to distance or direction changes [18]. Yu et al. presented a GR algorithm on the ground of continuous hidden Markov model and optical flow method. This method can achieve feature segmentation and extraction of dynamic gesture information, ensuring accurate recognition of dynamic gestures. Compared with the dynamic time warping and group optimization radial basis function network algorithm, this method has significant advantages in gesture identity authentication [19].

In summary, significant breakthrough was achieved in the research of GR and tracking related technologies, but there are still many gaps and shortcomings in the research of dynamic GR and gesture tracking technology. This includes optimization of image sequence feature extraction, gesture tracking combined with skin color detection, and precision and speed processing of complex environmental backgrounds. In response to these shortcomings, research has explored dynamic GR technology.

### III. DYNAMIC GR MODEL ON THE GROUND OF IMPROVED PSO AND KCF

Gesture tracking refers to continuously tracking and locating the position and motion of human hands in video or image sequences, usually tracking the position and motion trajectory of human hands during hand movements in dynamic environments. Gesture tracking is an important part of dynamic GR, involving computer vision and image processing technologies, including object detection, motion estimation, and visual tracking. For enhancing the accuracy and speed of dynamic GR, this study conducted research on three key links: image sequence feature extraction, gesture detection, and gesture tracking.

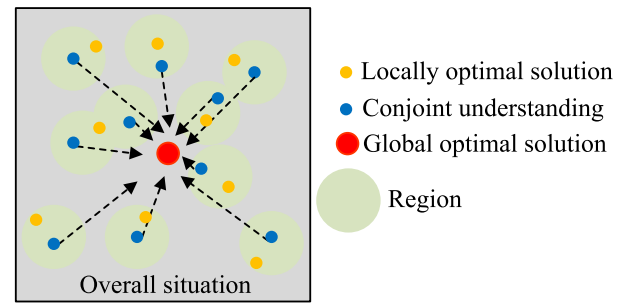


FIGURE 1. Schematic diagram of the working mechanism of PSO.

#### A. DESIGN OF AN OPTIMIZED MODEL FOR FEATURE EXTRACTION OF RGB IMAGE SEQUENCES ON THE GROUND OF IMPROVED PSO ALGORITHM

A monocular RGB camera can capture image information in three color channels: red, green, and blue, and its volume is small. It is commonly used in various devices such as smart mobility, smart homes, and HCI [20]. The research on dynamic GR is on the ground of images and video sequences captured by a monocular RGB camera. Feature extraction on the ground of RGB image sequences is an important way to ensure high-precision recognition of dynamic gestures. However, due to complex backgrounds, occlusion of gestures, and differences in hand shape and hand movements, the accuracy of feature extraction is poor [21]. In response to the problem of multiple image features and large feature dimensions, this study chose PSO for RGB image feature selection, reducing feature dimensions and achieving high-precision feature classification.

Data augmentation is a common data preprocessing method, and online data augmentation has been studied to enhance the original image data [22], [23]. It randomly extracts images at a ratio of 60% and modifies the resolution of RGB and corresponding depth images from  $640 \times 480$  to  $320 \times 560$ . It randomly extracts 20% of the images for mirror operation. It randomly extracts 10% of the images for small-scale rotation. It performs brightness enhancement and gamma transformation on 25% of the extracted images.

The growth in the number of image features can help enhance recognition accuracy, but it can cause feature combination dimensions to be too high or feature redundancy, leading to increased computational costs. This study introduces PSO algorithm for feature optimization. PSO is a heuristic optimization algorithm inspired by the foraging behavior of natural bird or fish schools. It simulates the behavior of bird or fish schools searching for food, constantly updating the position and velocity of particles to find the optimal solution [24], [25]. The PSO algorithm possesses the demerits of being simple to implement and not easily trapped in local optima, and is widely used to solve optimization problems. The working mechanism of PSO is shown in Figure 1.

The PSO algorithm treats the potential solution of the problem as particles, first randomly generating a certain

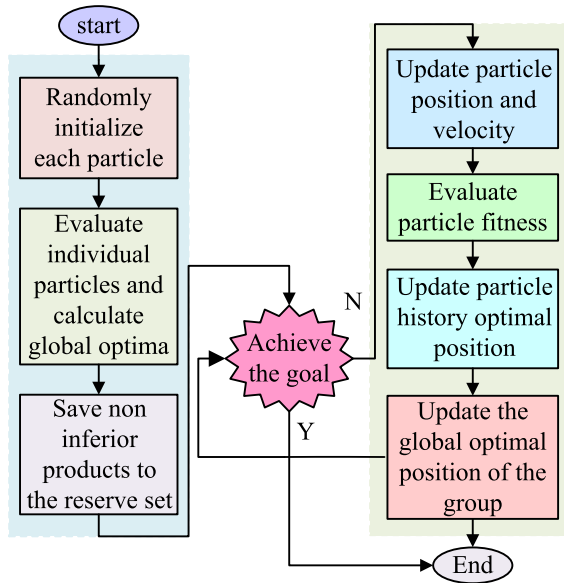


FIGURE 2. PSO algorithm process.

number of particles and randomly assigning initial positions and velocities to each particle. Particles adjust their position and velocity on the ground of their own experience and the experience of neighboring particles. The process of updating particle position  $x_i^m$  and velocity  $v_i^{m+1}$  is shown in equation (1), where  $w$  serves as the inertia weight.  $c_1$  and  $c_2$  serve as the acceleration factors for individuals and groups, respectively.  $r_1$  and  $r_2$  serve as a random number,  $\alpha$  represents a constraint factor, and  $m$  serves as the number of iterations.  $p_i$  serves as the optimal position of particles under local search.  $s$  serves as the optimal position of particles under global search.

$$\begin{cases} v_i^{m+1} = wv_i^m + c_1r_1(p_i^m - x_i^m) + c_2r_2(s^m - x_i^m) \\ x_i^{m+1} = x_i^m + \alpha v_i^{m+1} \end{cases} \quad (1)$$

The update of particle position is influenced by the particles themselves and neighboring particles. The particles communicate with each other to achieve collective search and information sharing, forming a particle swarm communication topology network. The complete PSO algorithm process is shown in Figure 2. The PSO algorithm also has some drawbacks, as it is more sensitive to parameter selection. The parameter selection determines the initial performance of the algorithm, and the generation of particle numbers, particle update speed, learning factor, and inertia constant affect the distribution of particles. This study restricts particle velocity on the ground of the search space of actual situations, and empirically takes values for learning factors and inertia constants.

This study introduces adaptive inertia weights on traditional PSO techniques to reduce the dimensionality of high-dimensional features, and optimizes feature combinations to complete image feature extraction. The expression for the adaptive inertia weight  $w_i^d$  is shown in equation (2),

where  $w_{\min}$  and  $w_{\max}$  represent the set minimum and maximum inertia coefficients, respectively.  $f_{\text{average}}^d$  represents the average fitness of particles at the  $d$ -th iteration.  $f(x_i^d)$  and  $f_{\min}^d$  represent the fitness and minimum fitness of the particle at the  $d$ -th iteration, respectively.

$$w_i^d = \begin{cases} w_{\min} + (w_{\max} - w_{\min}) \frac{f(x_i^d) - f_{\min}^d}{f_{\text{average}}^d - f_{\min}^d} f(x_i^d) & \geq f_{\text{average}}^d \\ w_{\max} f(x_i^d) & < f_{\text{average}}^d \end{cases} \quad (2)$$

Using a mixture of real and binary encoding to describe particle information, the particle's position vector  $X_i^R$  and velocity vector  $V_i^R$  are described in equation (3). In equation (3),  $f$  represents the feature index,  $f \in [1, F]$ .  $x_{if}^R \in [0, 1]$ ,  $v_{if}^R \in [-V_{\max}, V_{\max}]$ .

$$\begin{cases} X_i^R = (x_{i1}^R, x_{i2}^R, \dots, x_{iF}^R) \\ V_i^R = (v_{i1}^R, v_{i2}^R, \dots, v_{iF}^R) \end{cases} \quad (3)$$

In the process of feature optimization selection, the particle position is represented as binary encoding  $x_{if}^R$ , as shown in equation (4). When the position vector  $x_{if}^R$  is greater than 0.5 and  $x_{if}^B = 1$ , the features are included in a good feature combination.

$$X_i^B = (x_{i1}^B, x_{i2}^B, \dots, x_{iF}^B) \quad (4)$$

## B. GESTURE DETECTION MODEL ON THE GROUND OF SKIN COLOR DETECTION AND EXTREME LEARNING MACHINE

In human-computer interaction, the use of gestures can make the user more natural and intuitive when operating a device or application, and gesture detection can realize operations without contact, increasing the convenience of interaction. Multimodal interaction can also be realized through gesture detection, providing more options for interaction. Gesture detection plays an important role in human-computer interaction, and the research of gesture detection is very important.

Skin tone modeling is a technology used in computer vision and image processing to recognize and segment skin regions. Skin tone modeling an extensive range of utilization in areas such as facial recognition, human pose recognition, and nude detection [26], [27]. Usually, skin tone modeling converts the RGB color space to other color spaces to better capture the color features of the skin. Then it uses color thresholds to determine areas in the image that may contain skin, and by adjusting the range of color space thresholds, more accurate skin detection can be performed on people with different skin tones [28].

The common threshold skin color model is on the ground of the hexagonal cone model and Luminance, Chroma red, Chroma blue (YCrCb) color space. The hexagonal cone model, also known as the Hue, Saturation and Value (HSV)



color space, is expressed in Equation (5) for the skin color model on the ground of the HSV color space.

$$\begin{cases} (0^0 \leq H \leq 25^0) \cup (335^0 \leq H \leq 360^0) \\ (0.2 \leq S \leq 0.6) \cap (0.4 < V) \end{cases} \quad (5)$$

The expression of the YCrCb elliptical skin color model on the ground of the YCrCb color space is shown in equation (6), where  $\theta$  represents the inclination angle of the ellipse in the plane.  $a$  and  $b$  are the long and short half axes of the ellipse, respectively.  $e$  represents elliptical eccentricity.  $cx$  and  $cy$  represent the center of the ellipse.

$$\begin{cases} \frac{(x - ecx)^2}{a^2} + \frac{(y - cy)^2}{b^2} = 1 \\ \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} Cb - cx \\ Cr - cy \end{bmatrix} \end{cases} \quad (6)$$

Both threshold segmentation models use fixed thresholds to establish skin tone models, which lack adaptability to fluctuations in gesture skin tone caused by changes in environment and lighting. And all models are on the ground of a single color space, lacking mastery of skin tone patterns. The study introduces a Gaussian Mixture Module (GMM) to model HSV and YCrCb color spaces, and uses data training to summarize the Gaussian distribution patterns of the two color spaces. The GMM establishment process is shown in equation (7). In equation (7),  $P(x)$  and  $G$  represent the mixed Gaussian model and a single Gaussian distribution function, respectively.  $x$  represents gesture skin color sample data.  $\omega_k$  serves as the weight of the Gaussian component  $k$ .  $\mu_k$  and  $\sum_k$  represent the mean and variance of the  $k$ -th Gaussian component, respectively.

$$P(x) = \sum_{k=1}^K \omega_k \cdot G(x; \mu_k, \sum_k) \quad (7)$$

The offline training process of the GMM skin color model is shown in Figure 3. The skin tone training dataset consists of an offline skin tone database, an online collection of hand skin tone data, and a skin tone background database. After normalization, the dataset is transformed into HSV, YCrCb color spaces to form the training data and cross validation data feature sets. The GMM model training first uses K-means clustering algorithm to initialize the mean values of different Gaussian components of the GMM skin color model. Then it uses the Expectation Maximization (EM) method with hidden variables for estimating the parameters of the GMM skin color model, and obtains  $(x; \mu_k, \sum_k)$  under the Gaussian component.

The posterior probability  $\gamma_{jk}$  of the sample data generated by Gaussian mixture components in the EM algorithm is calculated using equation (8), where  $K$  serves as the number of Gaussian distributions. It recalculates the parameters  $x; \mu_k, \sum_k$  of  $K$  Gaussian distributions in the GMM model

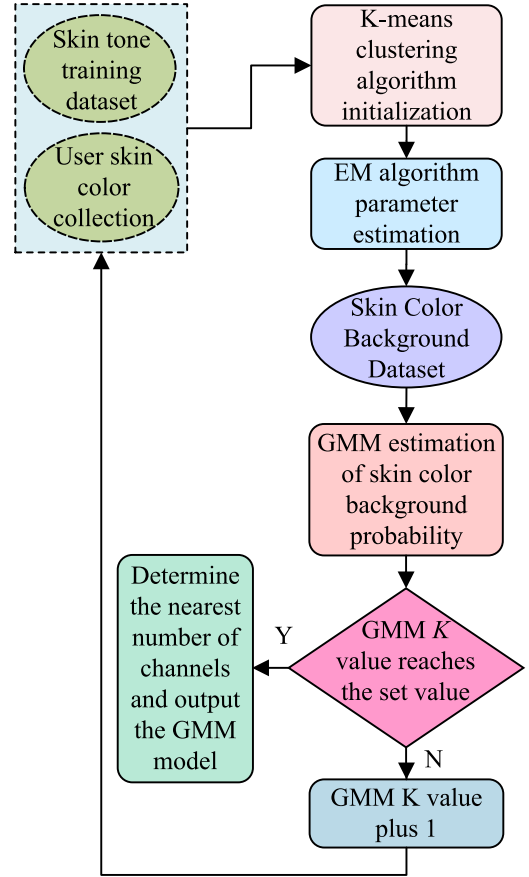


FIGURE 3. Schematic diagram of offline training process for gmm skin color model.

on the ground of  $\gamma_{jk}$ .

$$\gamma_{jk} = \frac{w_k G(x; \mu_k, \sum_k)}{\sum_{k=1}^K w_k G(x; \mu_k, \sum_k)} \quad (8)$$

Finally, the experimental GMM model calculates the average Gaussian response  $p$  in the cross validation data feature set, and uses the average Gaussian response to estimate the performance of the GMM model. The calculation process is showcased in equation (9). In equation (9),  $\alpha$  serves as the weight of the data feature set.  $N$  serves as the number of feature sets. It repeats the process continuously until the set maximum  $K$ -value is reached, and outputs the trained GMM model.

$$\begin{aligned} \text{Accuracy} &= \frac{\alpha_{skin}}{N_{skin}} \sum_{N_{skin}} p_{skin} - \frac{\alpha_{background}}{N_{background}} \sum_{N_{background}} p_{background} \end{aligned} \quad (9)$$

The GMM model outputs a Gaussian likelihood map normalized to a grayscale image, which is processed using the OSTU binarization method to obtain a binarized image. Then, the clutter and small contour areas in the image are removed

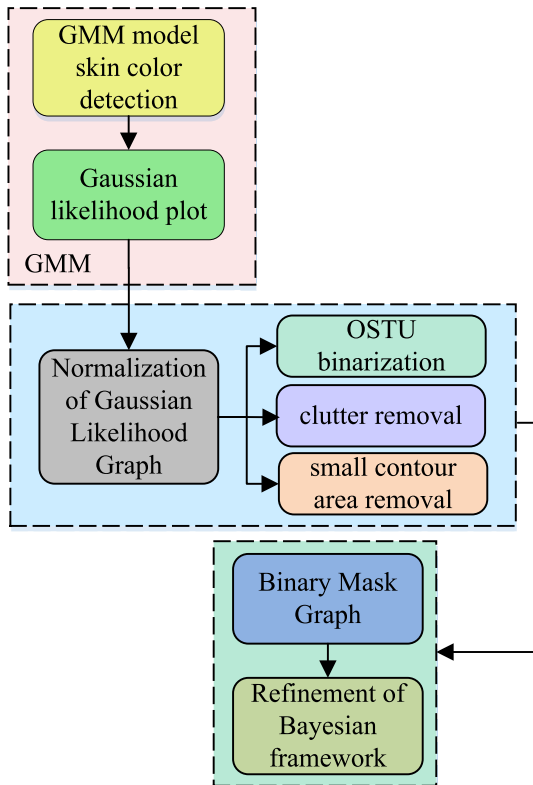


FIGURE 4. Bayesian framework refinement process.

to obtain a binary mask image of the skin color. Finally, the Bayesian framework was introduced to refine the skin color region of the image. The image processing process is shown in Figure 4.

This study combines the results of GMM skin color detection with Extreme Learning Machine (ELM) to complete GR and detection. ELM is a fast learning algorithm for single hidden layer feedforward neural networks. During the ELM training process, only the parameters from the input layer to the hidden layer need to be randomly initialized, and the parameters from the output layer to the hidden layer can be directly learned through analytical solutions. This avoids the process of iterative adjustment of connection weights in neural networks. The ELM classification is used to recognize the probabilities of different gesture regions belonging to different gesture categories in the gesture region images processed by the GMM model and Bayesian framework. The algorithm flow is shown in Figure 5.

### C. GESTURE TRACKING MODEL ON THE GROUND OF IMPROVED KCF ALGORITHM

Dynamic gesture tracking realizes the interaction with the computer system through the design of gestures and movements, which provides a more natural and intuitive interaction and can provide a more immersive user experience. At the same time, dynamic gesture tracking provides a faster and

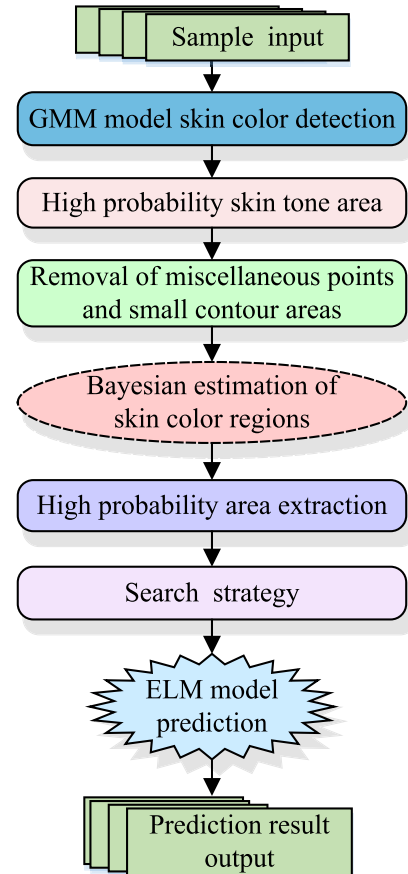


FIGURE 5. Dynamic gesture detection algorithm flowchart.

more direct way of operation and improves operational efficiency. Dynamic gesture detection and tracking is one of the important technologies in the field of human-computer interaction.

On the basis of completing gesture detection, tracking research was conducted on various changing gestures in monocular RGB image sequences. Kernel correlation filter is an image processing technique on the ground of kernel methods, which has the ability to model target features nonlinearly and is robust to scale and rotation changes. It is commonly used in tasks such as target tracking, image recognition, and image reconstruction [29]. KCF has good target tracking ability. After improving the PSO algorithm to extract features, research is conducted to determine gesture tracking results on the ground of the confidence results of the KCF algorithm. If the confidence is low, the GMM-ELM gesture detection results are used to initialize the KCF filter. The working principle is shown in Figure 6.

The essence of KCF is to solve the nonlinear ridge regression function, as showcased in equation (10). In equation (10),  $w$  represents the weight of the nonlinear ridge regression function.  $x_i$  and  $y_i$  represent the feature vectors and sample labels of the data samples, respectively.

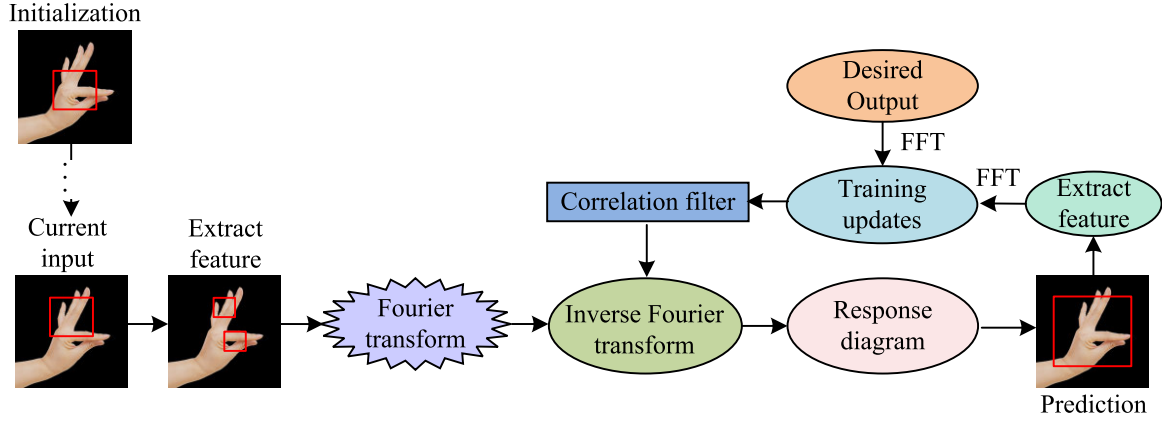


FIGURE 6. Working principle of improved KCF algorithm.

$\lambda$  represents a regular constraint.

$$\begin{cases} f(x) = w^T \phi(x) \\ \min_w \sum_i (f(x_i) - y_i) + \lambda \|w\|^2 \end{cases} \quad (10)$$

The process of estimating and solving the Gaussian response of image sequences for nonlinear ridge regression parameters is as follows: Firstly, the weights of the nonlinear ridge regression function are processed using Fourier transform, and the relevant process is showcased in equation (11). In equation (11),  $K^{xx}$  represents the kernel function.  $y$  represents the Gaussian likelihood map obtained from sample label processing.

$$\text{fft}(w) = \text{fft}(y) ./ (\text{fft}(K^{xx}) + \lambda) \quad (11)$$

Then it solves the Gaussian response of the tracking target, and the relevant process is showcased in equation (12). In equation (12),  $K^{xz}$  represents the kernel function,  $x$  and  $z$  represent the training sample feature matrix and test sample feature matrix, respectively.  $\text{ifft}$  represents the inverse Fourier transform.

$$\text{response} = \text{ifft}(\text{fft}(w)) .* \text{fft}(K^{xz}) \quad (12)$$

Finally, the kernel function is solved as showcased in equation (13).

$$K^{xx'} = \phi(\text{ifft}(\text{fft}(x) .* \text{fft}(x')))^T \quad (13)$$

Compared to Camshift filter and Kalman filter, KCF algorithm performs relatively well and has higher efficiency in real-time processing. However, the KCF algorithm still has some application shortcomings. When the gesture target undergoes non rigid changes, the KCF algorithm may not be able to adapt to the new scale of the target, leading to the failure of gesture target tracking. When the target is occluded or the image is noisy, the KCF algorithm is prone to interference, which can also lead to tracking loss of gesture targets [30]. Therefore, in dynamic GR, it is necessary to improve the KCF algorithm by combining

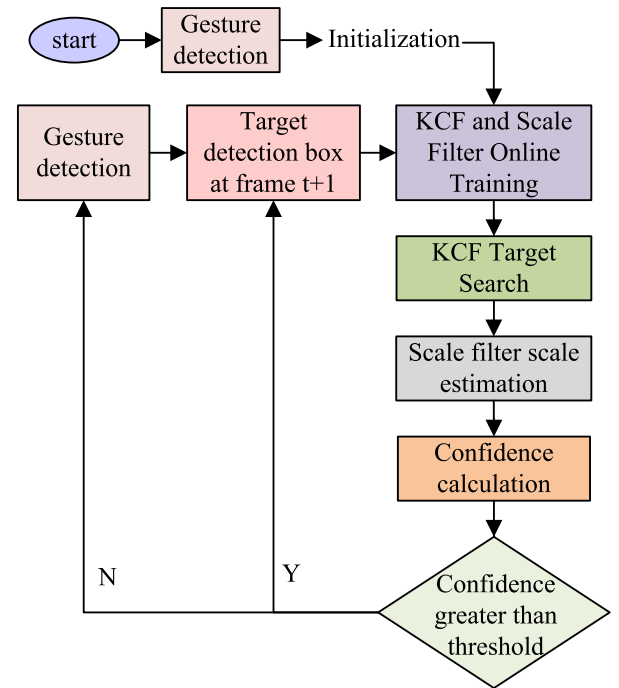


FIGURE 7. KCF algorithm improvement process.

other technologies to compensate for its shortcomings. The improvement process is shown in Figure 7.

This study introduces scale filters to improve the insensitivity of KCF algorithm to scale changes. The solution process of the optimal filter is shown in equation (14), where  $f$  represents the image block features.  $h$  represents scale filter.  $l$  represents the index of  $d$  image blocks extracted at different scales around the center of the gesture target in the previous frame image,  $l \in \{1, 2, \dots, d\}$ .  $g$  represents the one-dimensional Gaussian response function assigned to the image block.

$$\min \varepsilon = \left\| g - \sum_{l=1}^d h^l * f^l \right\|^2 + \lambda \sum_{l=1}^d \|h^l\|^2 \quad (14)$$

**TABLE 1.** Comparison of convergence of improved algorithms.

Test function	Algorithm	Average convergence value	Convergence times	Minimum number of iterations
Sphere	GA	8.212E-7	21	721
	PSO	3.353E-9	33	572
	Improve-PSO	5.202E-15	50	234
Schwefel	GA	8.282E-6	22	444
	PSO	2.222E-9	26	373
	Improve-PSO	6.332E-12	50	251
Rastrigin	GA	5.666E-6	19	476
	PSO	7.322E-9	28	373
	Improve-PSO	4.262E-14	50	264
Griewank	GA	6.438E-6	25	442
	PSO	1.453E-11	31	352
	Improve-PSO	2.423E-15	50	234
Ackley	GA	6.346E-6	22	448
	PSO	3.542E-9	27	348
	Improve-PSO	2.754E-12	50	229

After Fourier transform, the solution of the one-dimensional scale filter is shown in equation (15), where  $H$ ,  $G$ , and  $F$  represent the frequency responses of  $h$ ,  $g$ , and  $f$ , respectively.  $\chi$  represents the penalty factor.

$$H^l = \frac{\overline{GF}^L}{\sum_{k=1}^d \overline{F^k F^k} + \chi} \quad (15)$$

The scale filter uses the inverse Fourier transform to predict the target, and extracts fast features of images at different scales near the center of the gesture target for updating the scale filter. The updating process is showcased in equation (16). In equation (16),  $\eta$  serves as the parameter adjustment factor.  $A$  and  $A$  are undetermined parameters of the scale filter.  $t$  represents the frame rate of the image.

$$\begin{cases} A_t^l = (1 - \eta) A_{t-1}^l + \eta \overline{GF}_t^l \\ B_t = (1 - \eta) B_{t-1} + \eta \sum_{k=1}^d \overline{F_t^k F_t^k} \end{cases} \quad (16)$$

The confidence model for improving the KCF algorithm mainly consists of two parts. The first part is to determine whether the gesture target tracking has failed on the ground of the color histogram. The similarity calculation of the center area of the gesture target bounding box in the YCrCb color space between adjacent frames is showcased in equation (17). *hist1*, *hist2* represent color histogram feature vectors, respectively.

$$\text{similarity} = \frac{\text{hist1} \cdot \text{hist2}}{\|\text{hist1}\| \cdot \|\text{hist2}\|} \quad (17)$$

The second is to judge on the ground of the similarity of features between different image frames. The peak confidence *differVAL* is calculated according to equation (18), and *FM* represents the response peak.

$$\text{differVAL} = ((FM_{t-1} + FM_{t-2}) / 2 - FM_t) \quad (18)$$

#### IV. PERFORMANCE TESTING AND APPLICATION EFFECT ANALYSIS OF HCI GESTURE TRACKING MODEL

For testing the effectiveness of the feature extraction optimization algorithm, gesture detection, and tracking model

designed in the study, a series of performance testing experiments and dynamic gesture human-machine interaction effect analysis experiments were designed.

##### A. PERFORMANCE TESTING OF IMPROVED PSO FEATURE EXTRACTION OPTIMIZATION ALGORITHM

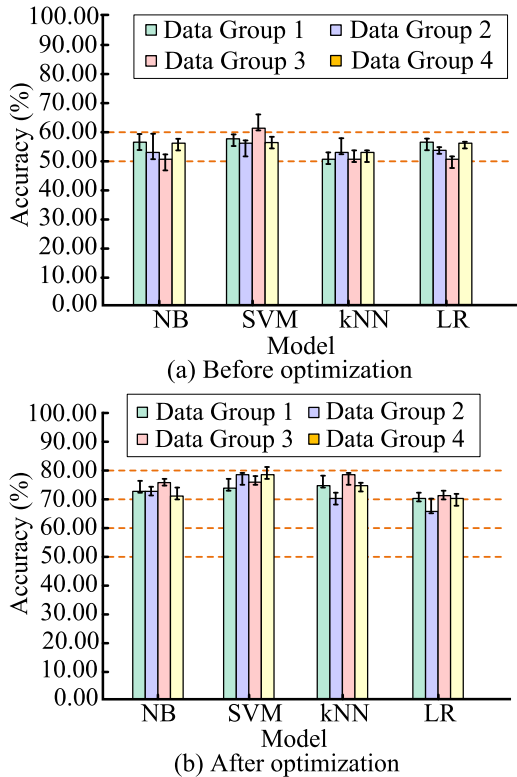
Firstly, this study designs a test experiment to analyze the optimization ability of the improved PSO algorithm. The study selected 5 sets of testing functions, namely single modal benchmark testing functions Sphere, Schwefel, multiple multimodal benchmark testing functions Rastrigin, Griewank, and Ackley. It uses genetic algorithm (GA), traditional PSO, and improved PSO algorithms to solve different test functions. This study conducted 50 independent experiments to analyze and evaluate the convergence results from three perspectives. This includes the average value of the convergence optimal solution in 50 independent experiments, the number of times the optimal solution was solved in 50 independent experiments, and the average number of iterations completed in the experiments. The relevant outcomes are showcased in Table 1. As shown in Table 1, the improved PSO algorithm has the best convergence in solving. In multiple independent experiments, the optimization values for solving different test functions have been minimized. The number of times GA and PSO search for the optimal solution is smaller than that of the improved PSO algorithm, and the convergence rate of the improved PSO algorithm reaches 100%. Comparing the iteration times of different methods, the improved PSO algorithm has no more than 300 iterations and has the fastest convergence speed.

Ten subjects were selected for the collection of five different gesture images, with a total of four sets of different image data collected, one of which was used as the test set. It selects common classification baseline models for GR, including naive Bayesian Classifier (NB), Support Vector Machine (SVM), k-nearest neighbor (kNN), and Logistic Regression (LR), and tests the feature optimization ability of the improved PSO algorithm on the ground of model classification accuracy. The relevant outcomes are showcased



**TABLE 2.** Repetitive experimental results of different model evaluation indicators.

Evaluating indicator		Mean Relative Error	R-squared	Precision	Recall	T/fps
Adaboost	Test	0.461	0.732	0.726	0.706	19.22
	Train	0.472	0.702	0.717	0.692	18.76
Faster R-CNN	Test	0.552	0.788	0.764	0.744	22.12
	Train	0.532	0.803	0.778	0.721	20.23
SSD	Test	0.419	0.742	0.757	0.782	25.34
	Train	0.422	0.692	0.792	0.762	24.22
GMM-ELM	Test	0.345	0.822	0.876	0.821	34.22
	Train	0.321	0.823	0.873	0.824	36.32



**FIGURE 8.** Comparison of classification accuracy before and after feature optimization.

in Figure 8. Figures 8 (a) and (b) show that after improving the PSO algorithm for feature extraction and optimization, the classification accuracy of different models has significantly improved compared to before optimization. In Figure 8 (a), the classification accuracy of different models in the four datasets is roughly distributed in the range of 50%-60%. In Figure 8 (b), the classification accuracy of different models in the four datasets has improved to within the range of 70%-80%. Improving the feature optimization operation of PSO algorithm contributes to the improvement of classification model accuracy.

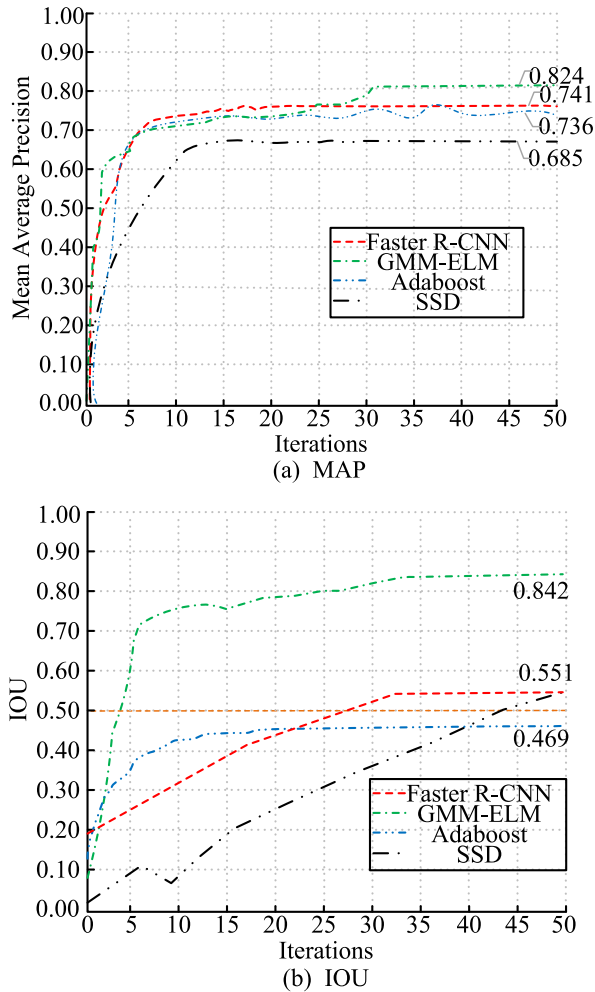
### B. PERFORMANCE TESTING OF GMM-ELM GESTURE DETECTION AND IMPROVED KCF GESTURE TRACKING MODEL

It selected NVGesture, HaGRID, Chalen LAP RGB-D Independent Gesture Dataset (Chalearn LAP RGB-D Isolated Gesture Dataset, Chalearn IsoGD), and MSRC-12 Kinect

Gesture Dataset as experimental datasets. The NVGesture dataset contains 1532 dynamic gestures, divided into 25 categories, mainly used for non-contact drivers. The HaGRID dataset was tested on subjects aged 18 to 65, and was collected under extreme conditions and strong lighting changes. Participants displayed gestures at a distance of 0.5 to 4 meters from the camera. ChalernIsoGD contains 47933 RGB-D gesture videos and 249 gesture tags. MSRC-12 contains 594 sequences and 719359 frames, collected by 30 people performing 12 gestures. It selects partial data from four different datasets as the experimental dataset and divides it into training and testing sets in an 8:2 ratio.

It compared the performance of Adaboost classifier, Faster R-CNN (Faster Regions with CNN Features), and SSD (Single Shot Multibox Detector) with GMM-ELM model. The relevant outcomes are showcased in Table 2. Table 2 showcases that the GMM-ELM model designed in the study has the optimal values for all indicators, and has good stability in both the test and training sets. The model has high repeatability. The minimum average relative error value of the GMM-ELM model is 0.321, which is 0.231 lower than the maximum value of 0.552 in the Faster R-CNN model. The accuracy and recall are significantly higher than other models, with values above 0.800, achieving the maximum balance between accuracy and recall. The correlation coefficient R-squared is a measure of the goodness of fit of a statistical model. The closer the value is to 1, the more excellent the model fits the data, that is, the stronger the explanatory power of the independent variable on the dependent variable. The correlation coefficient R-squared of the GMM-ELM model is closest to 1, indicating that the model has the best fit between the predicted value and the true value. The GMM-ELM model has a fast detection speed of 36.32 frames per second, while other models have detection rates below 30 frames per second. The smallest detection rate model is Adaboost.

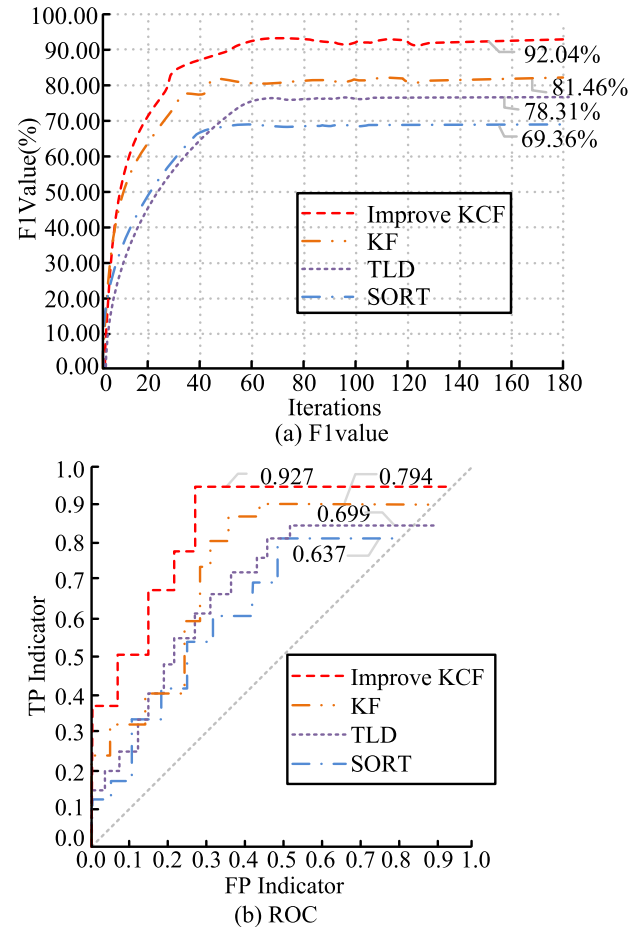
Mean Average Precision (mAP) and Intersection over Union (IoU) are commonly used evaluation indicators for object detection. The relevant outcomes are showcased in Figure 9. MAP is a comprehensive evaluation index for the performance of the GMM-ELM model for multi class object detection, while IoU can measure the accuracy of object detection algorithms. As shown in Figure 9, the mAP curve and IoU curve of the GMM-ELM model are both located at the top of the coordinate axis. The maximum mAP value in Figure 9 (a) converges to 0.824. The mAP of the SSD



**FIGURE 9.** Accuracy and intersection union ratio results of different detection models.

model converges to a minimum value of 0.685. However, due to the fact that the GMM-ELM model needs to be gradually optimized through random initialization parameters, the algorithm performance before 25 iterations is inferior to that of the Faster R-CNN model. The maximum IoU value of the GMM-ELM model in Figure 9 (b) converges to 0.842, showing significant differences relative to other models. The IoU values of the other three models are within the iteration range, and most of them are below 0.5, indicating poor accuracy in gesture target detection.

It compares the gesture tracking model Improved KCF designed for research with Kalman filtering (KF), SORT (Simple Online And Realtime Tracking), and TLD (Tracking Learning Detection). The relevant outcomes are showcased in Figure 10, using the F1 value of the model and the AUC (Area Under the Curve) under the Receiver Operating Characteristic Curve (ROC) as evaluation indicators. As shown in Figure 10 (a), the F1 value of the Improve KCF model converges to a maximum of 92.04%, while the SORT model converges to a minimum of 69.36%. The F1 value is a comprehensive evaluation index of classifier



**FIGURE 10.** Comparison of F1 value and ROC curve of gesture tracking model.

performance, which is the harmonic mean of accuracy and recall. It demonstrates that the overall performance of the Improved KCF model is superior to other target tracking models. As showcased in Figure 10 (b), the ROC curve of the Improve KCF model is located at the top left corner of the coordinate axis, with an AUC value of 0.927. The ROC curve is an indicator for evaluating the performance of binary classifiers, with AUC values ranging from 0 to 1, with larger values showcasing better classifier performance. Overall, the gesture tracking model designed in the study has significantly improved its tracking performance.

The average absolute error (MAE) and root mean square error (RMSE) results of different tracking models are shown in Figure 12. As showcased in Figure 12, the MAE and RMSE of the Improve KCF model both show a decreasing trend and converge to the minimum value. In Figure 12 (a), the minimum RMSE of the Improve KCF model is 0.07. In Figure 12 (b), the minimum MAE value of the Improve KCF model is 0.14. Both indicators represent the difference between predicted and actual values, and the smaller the value, the better the model fits the data. This indicates that the improve-KCF model can exhibit lower tracking errors in the tracking process of gesture targets.

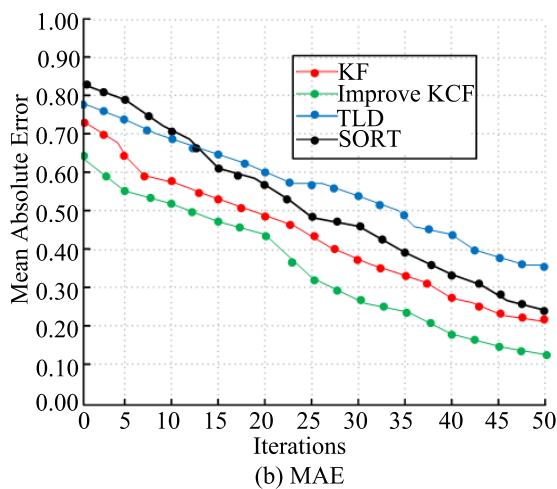
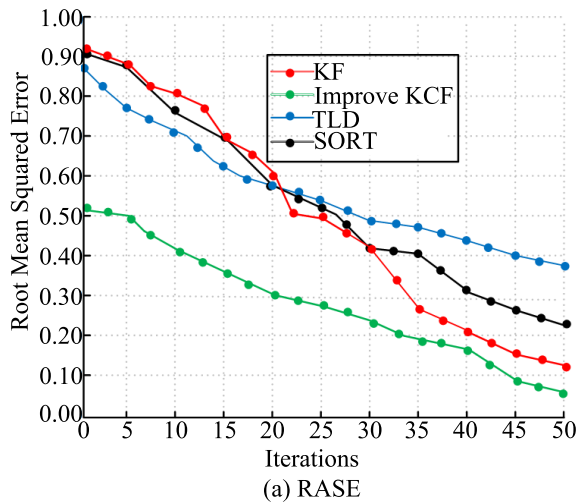


FIGURE 11. Comparison of error values of gesture tracking model.

### C. ANALYSIS OF THE APPLICATION EFFECT OF IMPROVING GESTURE DETECTION AND TRACKING ALGORITHMS

Under the operating system of 64bit and Windows 7, using the Visual Studio 2016 development toolkit, the algorithm for experimental design is written in C++ language. The camera of the interactive device uses a 2-megapixel RGB monocular camera. The interactive system can collect and annotate gesture data. The detection and tracking of gestures will be studied and improved algorithms and corresponding baseline models will be applied to interactive systems for comparative analysis.

Firstly, it analyzes the detection accuracy of gesture and background regions in the interaction system before and after improvement, as well as the rejection rate of gesture tracking. The accuracy of GR detection is related to the functionality, reliability, and user experience of the interaction system. Special attention should be paid to the accuracy of gesture detection when developing gesture interaction systems. Figure 12 (a) shows that compared with the application of the baseline model, the improved model's

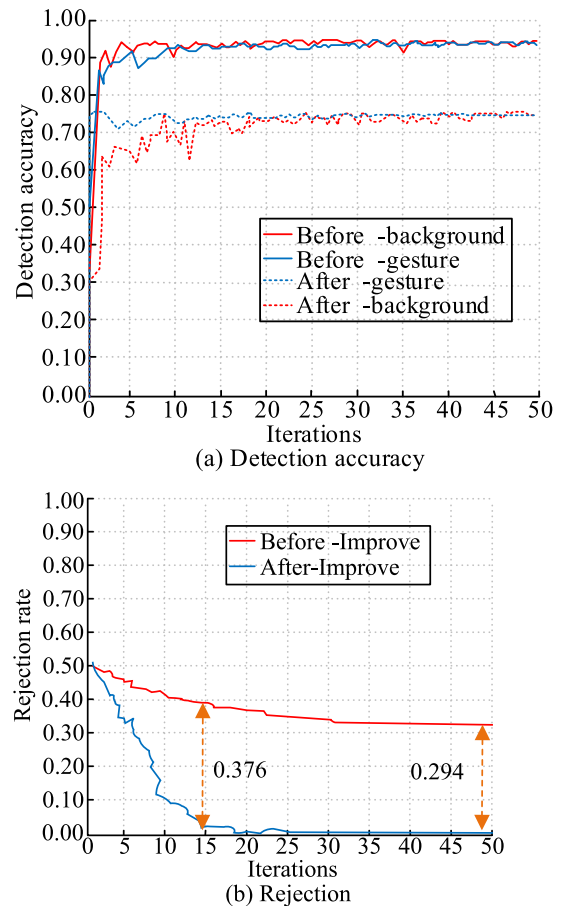


FIGURE 12. Comparison of HCI detection accuracy and rejection rate.

accuracy in detecting gestures and background regions in interactive systems increases to over 0.90 at the beginning of the iteration, and stabilizes around 0.954 after small fluctuations. The detection accuracy of the baseline model in the interactive system ultimately stabilizes around 0.746, and the accuracy fluctuates greatly during the iteration. The gesture interaction system has a certain degree of fault-tolerant ability to respond to user gesture changes and misoperations, correctly recognizing and providing feedback on user intentions. As shown in Figure 12 (b), the improved algorithm shows a significant decrease in the rejection rate in the interactive system, with a decrease level of 0.421. The rejection rate of the baseline model tends to stabilize after 15 iterations. The rejection rate of gesture tracking is the proportion of errors or unrecognizability that occur when the system recognizes user gestures. Overall, the research and design method has high accuracy in GR.

Finally, the application effect of the gesture tracking system in HCI devices was analyzed, and 30 volunteers who participated in the experiment were selected for evaluation. The age range of the participants ranged from 18 to 35 years old. The evaluation perspectives include interaction system stability, interaction diversity, scalability, agility, and overall user satisfaction. The experimental results are showcased

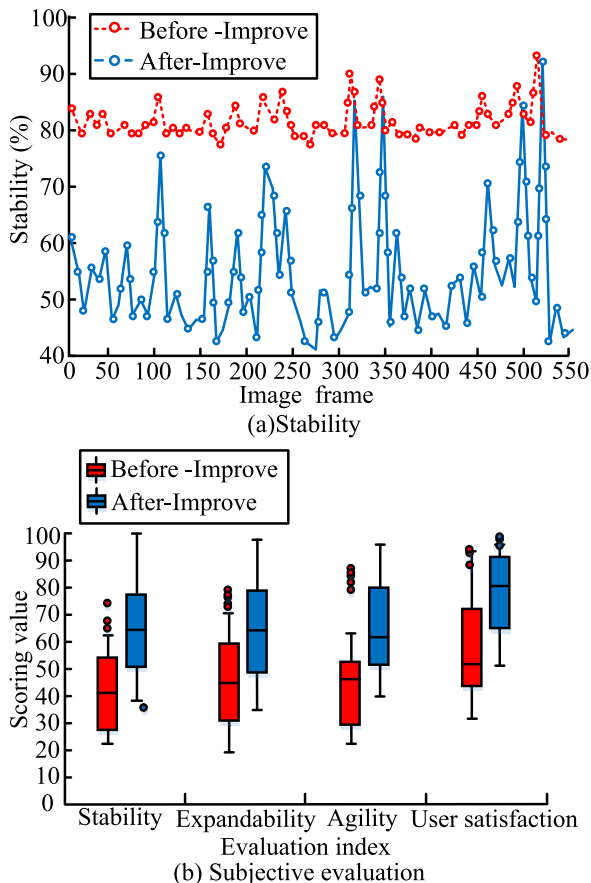


FIGURE 13. Application effect evaluation of HCI system.

in Figure 13. As showcased in Figure 13 (a), the method designed in the study has good recognition and tracking stability in interactive systems. Under the resistance of human interference, jitter, and other factors in the experiment, the stability fluctuates within a small range of 80% -90% as the image frame rate increases. The stability fluctuation range of the interaction system before improvement is relatively large, with a minimum stability of 41.06% and a maximum stability of 91.27%. As shown in Figure 13 (b), the improved interaction system has achieved significant improvement in all four different evaluation perspectives, with a significant increase in the mean score. Overall, after the research and design methods are put into use, the interaction system can recognize and track various types of gestures, flexibly expand and customize to meet user needs. And the system has a fast response speed, agility meets the real-time requirements of interaction, and overall user satisfaction and experience are good.

## V. CONCLUSION

Due to the interference of complex backgrounds and gesture changes, dynamic GR and tracking technology has been negatively affected to a certain extent. For enhancing the accuracy and tracking effect of dynamic GR, this study uses PSO algorithm for image sequence feature extraction.

Then, it introduced skin color model and kernel correlation filtering algorithm to detect and track long-term changing gestures under monocular RGB cameras, and constructed a gesture tracking model under dynamic GR. The experiment showcases that the improved particle swarm image sequence feature extraction method designed in the study has significantly improved the accuracy and speed of solving. After feature optimization, the classification accuracy of different classification models has been improved, with accuracy increasing from 50% -60% to within the range of 70% -80%. The average relative error, accuracy, recall, and correlation coefficient R-squared of the GMM-ELM model all reached their optimal values, with a detection speed of 36.32 frames per second and high repeatability on both the test and training sets. The mAP curve and IoU curve of the GMM-ELM model converge to the maximum values of 0.824 and 0.842, respectively, indicating better accuracy in gesture target detection. Compared with other target tracking models, the maximum F1 value of the Improved KCF model converges to 92.04%, the maximum AUC is 0.927, the minimum RMSE is 0.07, and the minimum MAE is 0.14. The tracking accuracy of the tracking model is better. In HCI systems, this method significantly improves the accuracy of gesture and background detection, as well as the rejection rate of targets. The stability of the interaction system is good, and the application evaluation is relatively good. This study helps to promote the application of HCI GR, but it did not focus on the real-time performance of gesture tracking, which can be used as a future research direction.

## REFERENCES

- [1] L. Guo, Z. Lu, and L. Yao, "Human-machine interaction sensing technology based on hand gesture recognition: A review," *IEEE Trans. Hum.-Mach. Syst.*, vol. 51, no. 4, pp. 300–309, Aug. 2021, doi: 10.1109/THMS.2021.3086003.
- [2] S. N. Amin, P. Shivakumara, T. X. Jun, K. Y. Chong, D. L. L. Zan, and R. Rahavendra, "An augmented reality-based approach for designing interactive food menu of restaurant using Android," *Artif. Intell. Appl.*, pp. 26–34, Oct. 2023, doi: 10.47852/bonviewaia2202354.
- [3] C. Liu and T. Szirányi, "Real-time human detection and gesture recognition for on-board UAV rescue," *Sensors*, vol. 21, no. 6, p. 2180, Mar. 2021, doi: 10.3390/s21062180.
- [4] L. Liu, W. Xu, Y. Ni, Z. Xu, B. Cui, J. Liu, H. Wei, and W. Xu, "Stretchable neuromorphic transistor that combines multisensing and information processing for epidermal gesture recognition," *ACS Nano*, vol. 16, no. 2, pp. 2282–2291, Jan. 2022, doi: 10.1021/acsnano.1c08482.
- [5] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, A. H. Alamoodi, O. S. Albahri, Z. T. Al-Qaysi, A. S. Albahri, and M. M. Salih, "Real-time sign language framework based on wearable device: Analysis of MSL, DataGlove, and gesture recognition," *Soft Comput.*, vol. 25, no. 16, pp. 11101–11122, May 2021, doi: 10.1007/s00500-021-05855-6.
- [6] D. Sengupta, J. Romano, and A. G. P. Kottapalli, "Electrospun bundled carbon nanofibers for skin-inspired tactile sensing, proprioception and gesture tracking applications," *npj Flexible Electron.*, vol. 5, no. 1, pp. 29–42, Oct. 2021, doi: 10.1038/s41528-021-00126-8.
- [7] Z. Song, Z. Cao, Z. Li, J. Wang, and Y. Liu, "Inertial motion tracking on mobile and wearable devices: Recent advancements and challenges," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 692–705, Oct. 2021, doi: 10.26599/TST.2021.9010017.
- [8] J. P. Sahoo, A. J. Prakash, P. Pławiak, and S. Samantray, "Real-time hand gesture recognition using fine-tuned convolutional neural network," *Sensors*, vol. 22, no. 3, p. 706, Jan. 2022, doi: 10.3390/s22030706.



- [9] A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, F. Burghardt, L. Benini, A. C. Arias, and J. M. Rabaey, “A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition,” *Nature Electron.*, vol. 4, no. 1, pp. 54–63, Dec. 2020, doi: [10.1038/s41928-020-00510-8](https://doi.org/10.1038/s41928-020-00510-8).
- [10] Y. Jiang, L. Song, J. Zhang, Y. Song, and M. Yan, “Multi-category gesture recognition modeling based on sEMG and IMU signals,” *Sensors*, vol. 22, no. 15, p. 5855, Aug. 2022, doi: [10.3390/s22155855](https://doi.org/10.3390/s22155855).
- [11] K. S. Yadav, A. Monsley K., and R. H. Laskar, “Gesture objects detection and tracking for virtual text entry keyboard interface,” *Multimedia Tools Appl.*, vol. 82, no. 4, pp. 5317–5342, Feb. 2023, doi: [10.1007/s11042-021-11874-0](https://doi.org/10.1007/s11042-021-11874-0).
- [12] W. K. Wong, F. H. Juwono, and B. T. T. Khoo, “Multi-features capacitive hand gesture recognition sensor: A machine learning approach,” *IEEE Sensors J.*, vol. 21, no. 6, pp. 8441–8450, Mar. 2021, doi: [10.1109/JSEN.2021.3049273](https://doi.org/10.1109/JSEN.2021.3049273).
- [13] H. Ansar, A. Jalal, M. Gochoo, and K. Kim, “Hand gesture recognition based on auto-landmark localization and reweighted genetic algorithm for healthcare muscle activities,” *Sustainability*, vol. 13, no. 5, p. 2961, Mar. 2021, doi: [10.3390/su13052961](https://doi.org/10.3390/su13052961).
- [14] Q. Gao, Y. Chen, Z. Ju, and Y. Liang, “Dynamic hand gesture recognition based on 3D hand pose estimation for human–robot interaction,” *IEEE Sensors J.*, vol. 22, no. 18, pp. 17421–17430, Sep. 2022, doi: [10.1109/JSEN.2021.3059685](https://doi.org/10.1109/JSEN.2021.3059685).
- [15] K. S. Yadav, K. A. Monsley, R. H. Laskar, S. Misra, M. K. Bhuyan, and T. Khan, “A selective region-based detection and tracking approach towards the recognition of dynamic bare hand gesture using deep neural network,” *Multimedia Syst.*, vol. 28, no. 3, pp. 861–879, Jan. 2022, doi: [10.1007/s00530-022-00890-1](https://doi.org/10.1007/s00530-022-00890-1).
- [16] E. Thabet, F. Khalid, P. S. Sulaiman, and R. Yaakob, “Algorithm of local features fusion and modified covariance-matrix technique for hand motion position estimation and hand gesture trajectory tracking approach,” *Multimedia Tools Appl.*, vol. 80, no. 4, pp. 5287–5318, Feb. 2021, doi: [10.1007/s11042-020-09903-5](https://doi.org/10.1007/s11042-020-09903-5).
- [17] S. Saboo and J. Singha, “Vision based two-level hand tracking system for dynamic hand gestures in indoor environment,” *Multimedia Tools Appl.*, vol. 80, no. 13, pp. 20579–20598, May 2021, doi: [10.1007/s11042-021-10669-7](https://doi.org/10.1007/s11042-021-10669-7).
- [18] A. Li, E. Bodanese, S. Poslad, T. Hou, K. Wu, and F. Luo, “A trajectory-based gesture recognition in smart homes based on the ultrawideband communication system,” *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22861–22873, Nov. 2022, doi: [10.1109/JIOT.2022.3185084](https://doi.org/10.1109/JIOT.2022.3185084).
- [19] P. Yu, J. Yin, Y. Sun, Z. Du, and N. Cao, “An identity authentication method for ubiquitous electric power Internet of Things based on dynamic gesture recognition,” *Int. J. Sensor Netw.*, vol. 35, no. 1, p. 57, Jan. 2021, doi: [10.1504/ijnsnet.2021.112889](https://doi.org/10.1504/ijnsnet.2021.112889).
- [20] S. Saboo, J. Singha, and R. H. Laskar, “Dynamic hand gesture recognition using combination of two-level tracker and trajectory-guided features,” *Multimedia Syst.*, vol. 28, no. 1, pp. 183–194, Feb. 2022, doi: [10.1007/s00530-021-00811-8](https://doi.org/10.1007/s00530-021-00811-8).
- [21] W. Juan, “Gesture recognition and information recommendation based on machine learning and virtual reality in distance education,” *J. Intell. Fuzzy Syst.*, vol. 40, no. 4, pp. 7509–7519, Apr. 2021, doi: [10.3233/jifs-189572](https://doi.org/10.3233/jifs-189572).
- [22] S. Ahmed, W. Kim, J. Park, and S. H. Cho, “Radar-based air-writing gesture recognition using a novel multistream CNN approach,” *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23869–23880, Dec. 2022, doi: [10.1109/JIOT.2022.3189395](https://doi.org/10.1109/JIOT.2022.3189395).
- [23] N. Mohamed, M. B. Mustafa, and N. Jomhari, “A review of the hand gesture recognition system: Current progress and future directions,” *IEEE Access*, vol. 9, pp. 157422–157436, 2021, doi: [10.1109/ACCESS.2021.3129650](https://doi.org/10.1109/ACCESS.2021.3129650).
- [24] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, “Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method,” *Appl. Nanosci.*, vol. 13, no. 2, pp. 1045–1054, Feb. 2023, doi: [10.1007/s13204-021-01970-w](https://doi.org/10.1007/s13204-021-01970-w).
- [25] J. Tian, M. Hou, H. Bian, and J. Li, “Variable surrogate model-based particle swarm optimization for high-dimensional expensive problems,” *Complex Intell. Syst.*, vol. 9, no. 4, pp. 3887–3935, Aug. 2023, doi: [10.1007/s40747-022-00910-7](https://doi.org/10.1007/s40747-022-00910-7).
- [26] J. Gangrade and J. Bharti, “Vision-based hand gesture recognition for Indian sign language using convolution neural network,” *IETE J. Res.*, vol. 69, no. 2, pp. 723–732, Feb. 2023, doi: [10.1080/03772063.2020.1838342](https://doi.org/10.1080/03772063.2020.1838342).
- [27] J. P. Sahoo, S. P. Sahoo, S. Ari, and S. K. Patra, “RBI-2RCNN: Residual block intensity feature using a two-stage residual convolutional neural network for static hand gesture recognition,” *Signal, Image Video Process.*, vol. 16, no. 8, pp. 2019–2027, Feb. 2022, doi: [10.1007/s11760-022-02163-w](https://doi.org/10.1007/s11760-022-02163-w).
- [28] R. Xia, Y. Chen, and B. Ren, “Improved anti-occlusion object tracking algorithm using unscented Rauch–Tung–Striebel smoother and kernel correlation filter,” *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6008–6018, Sep. 2022, doi: [10.1016/j.jksuci.2022.02.004](https://doi.org/10.1016/j.jksuci.2022.02.004).
- [29] J. Fan, X. Yang, R. Lu, W. Li, and Y. Huang, “Long-term visual tracking algorithm for UAVs based on kernel correlation filtering and SURF features,” *Vis. Comput.*, vol. 39, no. 1, pp. 319–333, Jan. 2023, doi: [10.1007/s00371-021-02331-y](https://doi.org/10.1007/s00371-021-02331-y).
- [30] L. Gao, B. Liu, P. Fu, M. Xu, and J. Li, “Visual tracking via dynamic saliency discriminative correlation filter,” *Int. J. Speech Technol.*, vol. 52, no. 6, pp. 5897–5911, Apr. 2022, doi: [10.1007/s10489-021-02260-2](https://doi.org/10.1007/s10489-021-02260-2).



**DINGHUA HE** was born in Hubei, China, in 1972. He received the B.S. degree from China University of Geosciences, Hubei, in 1995.

Since 1995, he has been a Teacher with Wuhan Polytechnic University, where he has been a Professor, since 2015. He is the author of five books and more than 12 articles. His research interests include algorithm and image recognition and harmony OS application development.



**YAN YANG** was born in Hubei, China, in 1984. She received the master's degree in system analysis and integration from Hubei University, in 2010.

From 2010 to 2021, she was a Teacher and the Director of the Software Engineering Department, Wuhan Qingchuan College. Since 2021, she has been a Teacher and an Associate Professor with Wuhan Polytechnic University. She is the author of two books and more than eight articles. Her main research interests include algorithms and information security.



**RANGZHONG WU** was born in Hubei, China, in 1971. He received the B.S. degree from China University of Geosciences, Hubei, in 1995, and the master's degree in engineering from Huazhong University of Science and Technology, in 2008.

Since 1995, he has been a Teacher with China University of Geosciences. He is the author of three books and more than eight articles. His research interests include algorithms and the Internet of Things.

...