Approach to hand posture recognition based on hand shape features for human–robot interaction

Jing Qi¹ · Kun Xu¹ · Xilun Ding¹

Received: 10 October 2020 / Accepted: 9 March 2021 / Published online: 10 April 2021
© The Author(s) 2021

Abstract

Hand segmentation is the initial step for hand posture recognition. To reduce the effect of variable illumination in hand segmentation step, a new CbCr-I component Gaussian mixture model (GMM) is proposed to detect the skin region. The hand region is selected as a region of interest from the image using the skin detection technique based on the presented CbCr-I component GMM and a new adaptive threshold. A new hand shape distribution feature described in polar coordinates is proposed to extract hand contour features to solve the false recognition problem in some shape-based methods and effectively recognize the hand posture in cases when different hand postures have the same number of outstretched fingers. A multiclass support vector machine classifier is utilized to recognize the hand posture. Experiments were carried out on our data set to verify the feasibility of the proposed method. The results showed the effectiveness of the proposed approach compared with other methods.

Keywords Human–robot interaction · Hand posture recognition · Skin color detection

Introduction

Human–robot interaction (HRI) based on a command line requires professionals to operate the robot, whereas HRI based on graphical user interface has brought non-expert users a lot of convenience. But these two methods do not meet the requirements of natural interaction and they have hindered the application of natural interaction between human and robots. To solve this problem, some researchers made a few attempts to introduce human communication into human–computer interaction [1]. Hand gestures can express much information and provide intuitive, natural, and effective interaction between human and robot. Hence, hand posture recognition and gesture-based interaction play significant roles in the human–robot interaction, and have attracted increasing attention [2].

Gesture recognition techniques can be divided into two categories: contact-based techniques and vision-based techniques [3,4], depending on whether physical interaction between the user and device occurs. Contact-based hand ges-

ture recognition technology is widely used for interactions via sensors such as electromyography, an inertial measurement unit, a data glove, multi-touch screen [3–5]. Although a contact-based system can obtain higher recognition rate and precision, users are required to wear the specific device while performing the hand gesture recognition. In turn, vision-based hand gesture recognition utilizes information captured by cameras such as monocular cameras, stereo cameras, color-depth (RGB-D) cameras, and it can provide a convenient and natural interface without requiring any wearable devices [2,3], and thus be widely used [2,6,7].

Although depth cameras have been used for several years in computer vision, the high price and poor-quality limit their application. With the release of low-cost RGB-D camera Kinect by Microsoft, the application of gesture recognition is broadened, because the RGB-D camera can provide high-quality depth images. The RGB-D cameras, such as Microsoft Kinect, and ASUS Xtion PROLIVE, can provide RGB, depth, and skeleton information. Some researchers use only depth [8] information, and some researchers use both RGB and skeletal information [9], and some researchers utilize both RGB and depth information [10,11] to recognize hand postures.

Generally, hand gestures are classified into static and dynamic gestures [12]. Hand shapes are static gestures, i.e.,

✉ Kun Xu
xk007@buaa.edu.cn

¹ Robotics Institute, School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

hand postures, while hand movements are dynamic gestures [13]. Generally, there are two steps to recognize a hand posture, i.e., hand segmentation and hand posture recognition, when conventional RGB cameras are used. Hand segmentation is the initial step in the process of hand posture/gesture recognition, where the image region with hands is detected and segmented from the image. Some specific features such as skin color [14–18], shape [19,20], both skin color and shape [21–23], motion [24], a combination of motion, skin color and edge [25] information are typically used for segmentation using conventional RGB cameras. The methods using motion information are optical flow [26], frame difference [27], background subtraction [28,29], etc. The optical flow method has a wide range of application. However, it is complex and hard to meet real time requirement. The frame difference method uses the difference between two consecutive images to detect moving objects [30]. The background subtraction method applies to static background scenes. And it has good real-time quality, but poor adaptability to abrupt changes of environment.

Due to the stable characteristics and translation invariance, skin color detection is frequently used to segment the hand region. Moreover, hand detection/segmentation that utilizes skin color information is easy to be implemented and requires little computation. But it is sensitive to variable illumination. In addition, the color of skin-like objects in the background is quite similar to hand, so hands can not be segmented accurately only using skin color information, when there is skin-like background. In this condition, skin information can be combined with other information, such as gradient, texture and histograms information, to detect hands accurately. The hand detection method using shape information usually requires to train a classifier using texture, histograms, edges, gradient information, etc. Thus, it is not sensitive to variable illumination. However, it usually involves higher computational complexity. The hand segmentation method using both skin color and shape information reduces the calculation quantity and improves the detection rate and reliability, comparing with the hand detection method using only shape information. However, it is influenced by variable illumination to some extent.

After the segmentation of hands from the image, hand postures/gestures will be recognized. Several methods were proposed to recognize hand postures/gestures. Some researchers used hidden Markov model (HMM) [31], dynamic time warping (DTW) [32] methods, etc. to recognize hand gestures. In addition, some researchers employed machine learning methods to recognize hand postures. Moreover, some researchers utilized histogram of oriented gradients (HOG) [33], local binary patterns (LBP) [34], HOG, means, variances and Haar [35] features, and AdaBoost classifiers [36–39] to classify hand postures. Some researchers used spatial histogram coding of nonsubsampling contourlet trans-

form coefficients [23], Gabor [40], local histogram [41], multiple kernels [42], saliency map, Gabor and pyramid histogram of oriented gradients [43], and support vector machine (SVM) [44,45] to classify hand postures. These existing methods consistently apply machine learning methods to train a model using the data which is prepared in advance. Then the trained model is utilized to recognize hand postures. Although it is required enough time to train the model, these methods are effective for classification once the model has been trained. Furthermore, some researchers utilized template matching algorithm [46–48] to recognize hand postures. Several studies that applied the geometric method [15,16,49–53] to recognize hand postures. These methods are simple and have no need of a pre-trained model. Moreover, based on edge detection, they often employ shape information for classification. However, edge detection is susceptible to noises, distortions, etc. Xiuhui Wang et al. [54] extracted the length and width of each finger, the number of fingers, the angle between wrist and each finger and the skin color as hand posture features to recognize hand postures using an extended genetic algorithm. Based their previous work, Yanqiu Liu et al. [55] added a set of concentric circles with centre at the centre of palm, then the number of concentric circles hitting the outlines of segmented hand region is extracted as hand posture feature. Next, a linear discriminant analysis algorithm was utilized to deal with these vectors. Finally, a weighted k-nearest neighbour algorithm is developed for hand posture recognition. Malima et al. [16] proposed an efficient method to recognize the hand posture by counting the number of zero-to-one (black-to-white) transitions in the circle with the centre at the centre of gravity (COG) of the hand region, and a radius at 70% of the farthest distance in the hand region from the COG. However, it is difficult to recognize the hand posture when different postures have the same number of stretched fingers. Ju et al. [21] divided hand posture into two categories: a fist and an open palm. And the open palm is recognize using the method of Malima et al. [16].

We found the work of Malima et al. [16] is much needed in our application scenario of interaction of human and our hexapod robot. However, this work is not well applied to our current application, so it is necessary to improve this work to make it suit our application scenario, this is the starting point of our work.

This work is motivated by the study of Malima et al. [16], where the authors classified hand postures with a different number of stretched fingers that denote different hand postures. However, they did not consider some hand postures with the same number of stretched fingers, that denote different hand postures. We focus mainly on the recognition of different hand postures that have the same number of the stretched fingers, and developed new hand shape distribution features to distinguish hand postures, particularly in the

cases where the same number of stretched fingers represents different hand postures.

To solve the problem and effectively recognize the hand posture, we developed a method of hand posture recognition using low-level edge features. Because segmented hand region is a closed area, polar coordinates are employed to describe the feature. The original point of the polar coordinate is the centroid of the hand region. After the features are extracted from the hand regions, the multiclass SVM classifier is employed to recognize the hand posture.

In addition, we focus on tasks of reconnaissance, rescue etc. in public security applications, based on characteristics of our integrated leg–arm hexapod robot. When reconnaissance tasks are performed, only hand postures are used to control the movements of the robot, because of the concealment of reconnaissance. Furthermore, robots can execute known tasks in a structured environment. However, for unknown tasks in an unstructured environment, which is often required by public security personnel, cannot be easily fulfilled. In this case, human knowledge and experiences are utilized, a supervised pattern and demonstration pattern are combined, and a method of linkage of movement and manipulation of robots is proposed based on hand posture recognition.

Our robot has multiple ways of movement, such as “3 + 3” gait, “2 + 4” gait, and “1 + 5” gait. Additionally, the robot has two manipulators, and different tools can be installed in the end effectors of the manipulators, such as a clamp and scissors. Thus, it requires multiple interactivities. Moreover, it needs natural interaction between a human and the robot. The difficulty of linkage of multiple movements and manipulations of the robot is how to design an interaction system based on hand posture recognition, so that public security personnel can perform tasks conveniently, naturally and efficiently. To solve the problem, we combine a supervision pattern and demonstration pattern, and propose an interactive method to achieve tasks of reconnaissance, rescue, etc. Specifically, tasks that require a robot to perform are mainly divided into two categories: regular tasks and complex tasks. Regular tasks are achieved by robots, such as move forward, backward, left and right. While complex tasks, such as unknown tasks in an unstructured environment, cannot be accomplished by the robot autonomously. A good solution is learning from demonstration. For example, hand postures are utilized to control movements of the robot. In addition, videos captured by the camera installed on the robot is used to supervise the operations of the robot, to make certain that the robot perform the desired action.

The main contributions of the proposed method are listed as follows:

1. To classify some hand postures that have the same number of stretched fingers, while representing different hand postures, this study presents a new hand shape distribu-

tion feature, motivated by the study of Malima et al. [16]. The experimental results demonstrate the effectiveness of the proposed feature.

2. To reduce the effect of illumination, a new CbCr-I component Gaussian mixture model (GMM) is developed to detect skin regions in this case. Furthermore, a new adaptive threshold is presented to reduce false detection and misdetection of skin pixels. In addition, a hand segmentation method using shape and position information is presented based on the CbCr-I component GMM. The segmentation results demonstrate the effectiveness of the CbCr-I component GMM, the adaptive threshold and the proposed segmentation method.

The remainder of this paper is organized as follows: the next section describes the proposed approach in detail, including hand segmentation, hand shape distribution feature extraction and posture recognition. The subsequent section provides the related experiments and results. In the final section, the conclusions of our work are summarized.

Proposed approach

Users are usually not experts in robotics, and natural interactions between human and robots are required. Because the reconnaissance task has features of concealment and dangerous, it requires no sound for interaction. And hand postures are natural, intuitive and non-verbal, so they are chosen for reconnaissance and rescue tasks in this case. In addition, knowledge gained and learned from humans is transferred to the proposed hand posture system to enhance HRI. Specifically, hand postures are regarded as graphics, and their maps are regarded as knowledge representations. Moreover, hand postures are used to control the movements/manipulations of the leg–arm hexapod robot in this study.

The proposed interaction system based on hand posture recognition is designed to enable the robot to perform reconnaissance, rescue, and counterterrorism tasks. First of all, several types of hand posture were designed according to daily communication-related actions, based on the requirements of the tasks and the characteristics of our robot. Furthermore, a mapping is predefined from a hand posture to the corresponding motion/manipulation of the robot. Specifically, ten kinds of hand postures are designed, and five kinds of hand postures are used to control movements of the robot, i.e., moving forward, backward, left, right and stop. While the remaining five types of hand postures are utilized to control manipulations of the robot, i.e., opening scissors, closing scissors, opening a clamp, closing a clamp, changing positions of a manipulator. And part of the mapping is shown in Table 1. Once the robot stops, and the hand posture which can change positions of a manipulator is recognized, the hand

Table 1 Mapping movements/manipulations of the hexapod robot

Types	1	2	3	4
Movements	Open scissors	Close a clamp	Change positions of a manipulator	Move forward

Fig. 1 Framework of proposed approach **Fig. 2** Framework of proposed hand segmentation approach

postures which control movements of the robot are utilized to control movements of a manipulator, to make the manipulator reach a specific position. Then images of hand postures were captured to form our data set. And proposed method of hand posture recognition is used to train our model to recognize the hand postures.

The proposed method of hand posture recognition consists of two stages, i.e. the training stage and the testing stage, which are illustrated in the flowchart of Fig. 1. Hand regions are segmented in the training stage, after skin regions are detected based on the CbCr-I component GMM. To distinguish between the face and hand, prior knowledge on the position is used to segment the hand region. The shape features are extracted from the hand region and serve as input vectors to build a multiclass SVM classifier model in the training stage, which is used to recognize hand posture in the testing stage.

Hand segmentation based on the CbCr-I component Gaussian mixture model

The hand region is segmented from the image in the hand segmentation step, which is the initial also essential step for hand posture recognition. The flow diagram of hand seg-

mentation is shown in Fig. 2. First of all, the input image was pre-processed. The pre-processing consists of bilateral filtering and transformation from normalized RGB to YCbCr and YIQ color spaces. Bilateral filtering is applied for remaining edge and denoising. Subsequently, skin regions are detected using the proposed CbCr-I component GMM. Afterwards, the number of skin connected components at the bottom of the image is calculated. If this is equal to zero, the hand was segmented using the long sleeves method. In contrast, if the number of skin connected components at the bottom of the image was greater than zero, the hand was segmented applying the short sleeves method.

Skin detection based on the CbCr-I component GMM

The primary step to detect skin color regions from the image is to choose a color space. To reduce the dependence on lighting, normalized RGB color spaces are applied [56]. The YCbCr color space can be used to detect skin due to its discreteness and clear separation of chrominance and luminance components [57]. Moreover, among the components of YIQ color space, the I-channel of the YIQ color space involves colors from orange to cyan, which makes it possible to detect skin pixels [58]. To detect skin color pixels more accurately,

both Cb and Cr components of the YCbCr color space and the I-component of the YIQ color space are utilized in this study to establish the skin color model.

A Gaussian mixture probability density function is the sum of weighted individual Gaussian probability density functions, and thus GMM is more capable of approximating a complex distribution. When the user interacts with the robot using hand postures, the scene, position and background are unknown. To detect the skin areas more accurately, a new CbCr-I component GMM is proposed in this case, which is expressed as follows:

$$P(X|\Theta) = \sum_{i=1}^k \alpha_i N_i(X|\mu_i, \Sigma_i) \quad (1)$$

$$\sum_{i=1}^k \alpha_i = 1, 0 \leq \alpha_i \leq 1 \quad (2)$$

$$\Theta = \{\alpha_1, \dots, \alpha_k, \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k\} \quad (3)$$

$$X = \{x_1, \dots, x_N\}, \quad (4)$$

where $N_i(X|\mu_i, \Sigma_i)$ denotes the i th Gaussian function, and μ_i, Σ_i are the mean and covariance matrix of the i th Gaussian function, respectively. The parameters of GMM are defined as Θ , and k is the number of Gaussian probability density functions; α_i is the weight of the i th Gaussian probability density function. X is the set of the pixel value; N is the number of pixels in the image; x_j ($1 \leq j \leq N$) is the j th pixel value of the Cb, Cr, and I components, which are obtained by means of converting from the normalized RGB color space into YCbCr and YIQ color spaces.

The expectation–maximization algorithm is adopted to estimate GMM parameters, and initial parameters of GMM are calculated by the k-means clustering technology. CbCr-I component GMM is established using the skin regions manually cut from hand posture images in our data set. To reduce calculation time, three Gaussian components were selected in this study.

The skin region detection steps are shown in Fig. 2. First of all, the input image was pre-processed. Several pre-processing methods were proposed, such as mean filtering, median filtering, and bilateral filtering. Sebastián Salazar-Colores et al. [59] presented an effective single image dehazing method by modifying the dark channel prior, which greatly reduces the recurrent artifacts presented using the ordinary dark channel. In this work, pre-processing consists of bilateral filtering and transformation from normalized RGB to YCbCr and YIQ color spaces. Bilateral filtering is applied for remaining edge and denoising. The probability that each pixel belongs to a skin pixel is calculated using the trained CbCr-I component GMM. A fixed threshold may cause false detection or misdetection, as a skin pixel value

is influenced by illuminance, background, etc. To solve this problem, a new adaptive threshold is presented in this case.

The process of skin detection by humans motivated us in this study. When a human detects skin regions from an unknown image, the overall situation of the image is evaluated, and skin regions are detected in detail, as our intuition. We use this process of skin detection by humans as a reference to set the adaptive threshold. Because the mean and middle values of the image and the difference between the high and mean values of the image can describe the overall condition of the image to some extent, the adaptive threshold is defined as Eq. (5).

$$t_a = at_\mu + bt_m + ct_s, \quad (5)$$

where t_a is the adaptive threshold, t_μ is mean value of the image, t_m is the middle value of the image, t_s is the difference between the high and mean values of the image, and a, b, c are empirical coefficients.

The probability that each pixel belongs to a skin pixel is calculated using the trained CbCr-I component GMM, then the adaptive threshold is applied to decide whether a pixel belongs to a skin pixel. In addition, morphological operations are applied, and skin regions are detected. When a human interacts with the robot using hand postures, neither the face nor the hand is on the left, right, top and bottom edge of the image, which is an assumption we make in our study. If there are connected components on the top, left, or right edge of the image, these will be removed.

Hand region segmentation

After the detection of skin regions, the hand region is segmented from the image. We assumed that when the user is gesturing, one hand is raised to gesture maintaining, the hand at a lower position than the top of the head, whereas the other hand is drooped. Moreover, it assumed that the captured image includes both the face and hand, and the position of the face is higher than the position of the gesturing hand, which is above the bottom edge of the image. When the user wears long sleeves, the number of skin regions on the bottom edge is equal to zero, while the number of skin regions on the bottom edge is greater than zero when the user wears short sleeves. When the user wears long sleeves, the hand region is segmented using the relative position information of the face and hand, i.e., the hand being lower than the face. Based on this assumption, the model automatically obtains the second lower skin area as the detected hand region. When the user wears short sleeves, the gesturing arm is detected using the prior position information, and the hand is segmented at the wrist using its shape feature.

Although there are various hand postures because of various hand joints, the shape of the arm remains basically

unchanged. Branch points are present on the palm, whereas there are no branch points on the arm. Moreover, the thickness of the forearm from the elbow to wrist becomes generally gradually smaller, and the thickness of the hand from wrist to palm widens suddenly. We can detect the wrist by shape information, and thus the hand region is segmented from the position of the wrist. The hand segmentation method is described in Algorithm 1.

Algorithm 1 Hand Segmentation based on the CbCr-I component GMM.

```

1: Skin regions are detected using the proposed CbCr-I component GMM.
2: The number of skin connected components at the bottom of the image is SkinBotNum.
3: if SkinBotNum = 0 then
4:   The hand region is segmented using the relative position information of the face and hand.
5: else
6:   if SkinBotNum > 0 then
7:     The gesturing arm can be detected using prior position information.
8:   To smooth the edges, erosion operation is utilized, followed by the image skeleton.
9:   NumBr is the number of branch points on the center line of the gesturing arm; NumBrH is the number of the branch points, which are higher than mean value of the resulting image skeleton; NumBrL is the number of the branch points, which are lower than the mean value of the center line of the gesturing arm.
10:  if NumBr = 0 then
11:    The whole curve is depicted by the center line of forearm.
12:  else
13:    if NumBr = NumBrH then
14:      The curve between the lowest branch points and the lowest point of the center line of the gesturing arm depicts the center line of the forearm.
15:    end if
16:    if NumBr = NumBrL then
17:      The curve between the highest points of the center line of the gesturing arm and the highest branch points is the center line of the forearm.
18:    end if
19:    if NumBr > NumBrH > 0, NumBr > NumBrL > 0 then
20:      The curve between the lowest branch points, which are higher than the mean value of all branch points, and the highest branch points, which are lower than the mean value of all branch points, is the center line of the forearm.
21:    end if

```

```

22:  end if
23:  To reduce the calculation capacity, the lower part of the center line of the forearm is removed, because the wrist is on the top of the center line of the forearm.
24:  A horizontal line is drawn for each pixel of the center line of the forearm, and the maximal horizontal distance between the intersection points of the horizontal line and the hand contour is defined as the horizontal distance.
25:  The point G is on the hand contour;  $g_G^a$  is the  $m$  gradient of the horizontal distance after the point G;  $g_G^b$  is the  $m$  gradient of the horizontal distance before the point G.
26:  if  $g_G^b \leq 0$ ,  $g_G^a \geq 0$  then
27:    The position of the maximal value of the gradient at G is defined as A, which denotes the possible location of the wrist.
28:  end if
29:  The position of the minimal distance below the position G is expressed as B.
30:  The contour between the points A and B depicts the approximate location of the wrist.
31:  The curvature of the points on the contour of the approximate wrist is calculated, and the position corresponding to the maximal curvature is the accurate position of the wrist.
32:  The hand region is segmented from the accurate position of the wrist.
33: end if
34: end if

```

Hand shape distribution feature

Motivation

Some researchers employed HOG [45], HOG and LBP [38], SURF and Hu moment invariant features [60] as features to recognize hand postures. These features commonly have higher computational complexity, and consume considerable CPU time, as these methods involve significant redundant feature information. The hand shapes provide semantic meanings of hand postures, and contours of hands are essential for the hand posture representation. Each contour of the hand posture represents only one hand posture class. This means that hand contours can clearly represent hand postures. As shown in Fig. 3, hand postures can be recognized with only hand contours, regardless of other information, such as texture and gradient. Furthermore, pixel-level features, such as pixel values, texture and gradient, need to be calculated at each pixel of the image. Thus, they require significant computation power. In this study, hand contour feature is adopted to recognize the hand posture, as the hand

Fig. 3 Examples of hand postures. (a) Hand postures with color, texture, gradient information, etc. (b) Hand postures with only hand contour information

contour is a common feature that can be calculated without requiring other information.

This work was motivated by the study of Malima et al. [16], and presents a new hand shape feature utilizing hand contour information to distinguish hand postures, especially when some hand postures have the same number of stretched fingers, whilst they represent different hand postures. Hence, the main problem in this case is the recognition of different hand postures, which have the same number of stretched fingers.

Comparing with some drawing circles methods [16] previously published, innovations of this case are listed in the following:

1. Drawing circles method [16] uses mainly distance or/and angle as features. Meanwhile, this work is motivated by shape context descriptor. The centroid of the segmented hand region is seen as the pole in polar coordinates, and we utilize the main direction of the segmented hand region as the reference direction. Moreover, a ray from the pole in the reference direction is seen as the polar axis. Furthermore, this work uses two-dimensional matrices which relates to shape context of hand contours as features.
2. Drawing circles method [16] classified hand postures when a different number of stretched fingers denote different hand postures. They did not address some hand postures that have the same number of stretched fingers, while representing different hand postures. This study mainly focuses on the latter cases, developing new hand shape features.
3. The features proposed in this study are different from the features in drawing circles method [16]. Drawing circles method [16] tracked the constructed the binary circle. Thereby, they used the pixel value of the constructed circle as features. In this study, hand contour information is adopted, and the distances between the points on the contour and the centroid are calculated, and one two-dimensional matrix which relates to the shape context of the hand contour is used to denote the feature of the hand shape.

4. Drawing circles method [16] calculated the number of zero-to-one (black-to-white) transitions, and then the number minus one (for the wrist) produced the number of stretched fingers of the hand postures. In this study, a multiclass SVM classifier is applied to recognize the hand posture.

Definition of the hand shape distribution feature (HSDF)

Hand contours can represent hand postures, thus the hand shape distribution feature (HSDF) is introduced to recognize hand postures in this study, which is shown as Algorithm 2.

Algorithm 2 Extract hand shape distribution feature from the hand region.

- 1: The skin region of the segmented hand is detected, and the segmented hand region and detected skin region within are normalized in size.
- 2: To smooth the edges of the detected skin region of the segmented hand region, erosion operation is applied.
- 3: Edges of the skin region of the segmented hand are detected using the Canny detector. Then the small areas are removed, and the skeleton operation is applied. Thus, a closed contour is obtained.
- 4: The pixel coordinates of the points on the contour are extracted using the method of the hand contour tracing based on eight-connected neighbourhood.
- 5: The centroid of the hand region is calculated using Eqs. (6)–(9), which is denoted as point O .
- 6: N is the number of points on the hand contour.
- 7: **for** $i = 1 \rightarrow N$ **do**
- 8: The distances between the centroid and the points on the hand contour are calculated by Eq. (10).
- 9: **end for**
- 10: The maximum value of the distances is denoted as L_{max} , and the corresponding point on the hand contour is point A .
- 11: The point O is seen as the pole in polar coordinates, the main direction of the hand region is the direction that the point O goes to the point A , and the ray from the pole in the main direction of the hand region is the polar axis.

12: **for** $j = 1 \rightarrow N$ **do**
 13: The angles between the main direction of hand posture and the points on the hand contour are calculated.
 14: **end for**
 15: L_{max} is divided into s equal parts, and length of each part is b . Then, a series of concentric circles are drawn, and the circle has a radius of kb , $k = 1, 2, \dots, s$. Furthermore, the angle of the preferred polar coordinate system is divided into t equal parts, so, $s \times t$ bins are obtained in the preferred polar coordinate system.
 16: **for** $m = 1 \rightarrow N$ **do**
 17: Judge the points on the hand contour belong to which bins.
 18: **end for**
 19: Calculate number of the points on the hand contour in each bin, then it is normalized, i.e. it is divided by total number of points on the hand contour N .

Hand regions are segmented using the skin color model mentioned above, and the segmented hand region and the detected skin region are shown in Fig. 4a and b, respectively. The edges of the segmented hand region are detected using the Canny detector [61]. The contours of the hand are closed curves, as shown in Fig. 4c. Generally, Cartesian rectangular coordinates are not suitable to represent the closed curve. Polar coordinates have natural advantages for representing the closed curve. Moreover, after the size of samples is normalized based on the length of features, the rotation invariant feature can be calculated in polar coordinates. Therefore, the features are described in polar coordinates.

We adopt the radius as a parameter to represent the contour of the hand. To calculate the radius of each point on the contour, the original point on hand regions should be set first. In our case, the centroid of the segmented hand region is adopted as the original point, because this point is located on the palm of hand, and it is hence almost at the centre of the hand. The centroid of the hand region is denoted as x_0, y_0 , and it is given by Eqs. (6)–(9).

$$x_0 = \frac{M_{10}}{M_{00}}, y_0 = \frac{M_{01}}{M_{00}} \quad (6)$$

$$M_{00} = \sum_{i=1}^N \sum_{j=1}^M P(i, j) \quad (7)$$

$$M_{10} = \sum_{i=1}^N \sum_{j=1}^M i * P(i, j) \quad (8)$$

$$M_{01} = \sum_{i=1}^N \sum_{j=1}^M j * P(i, j), \quad (9)$$

where $P(i, j)$ is the pixel value of the pixel (i, j) .

The red point in Fig. 4c is the centroid of the hand region, which is almost at the centre of the hand. The light blue curve depicts the contour of the hand region, and it can be seen clearly that the contour can roughly represents the hand posture. The distances between the centroid and the points on the hand contour are calculated as the radius of the points on the contour by Eq. (10).

$$R_{x,y} = \sqrt{(x_0 - x)^2 + (y_0 - y)^2}. \quad (10)$$

First, the distances between the centroid and the points on the contour are calculated clockwise. As shown in Fig. 4, the profile of the curve in Fig. 4d has a roughly corresponding relationship with the contour of hand regions in Fig. 4c. Meanwhile, the crests of curve depict the fingertips of contour of hand regions, as shown in Fig. 4d.

Selection of the origin

The selection of the origin is important to describe the features in our method, which can affect the representation of features of the hand contour to a large extent. However, our proposed method is robust with respect to the origin selection as long as the selected origin is on the palm.

Figure 5 shows the four classes of origin and the corresponding radius curves. The number of peaks and their location in the curves does not change, even when the locations of the original points are different. Hence, the features of the hand posture do not change with origin locations, which means the proposed feature extraction method for hand posture recognition is robust with respect to changes of the origin. This is reflected in the same tendency of the hand contour, where the feature value and location of fingertips remained at the local maxima, i.e., the number of peaks and their location in the curves remained unchanged.

Posture recognition

To deal with small-sample classification problem, the multi-class SVM classifier is employed to recognize the ten classes of hand postures. The proposed hand posture recognition method involves two stages, i.e., training stage and testing stage. In the training stage, skin color regions are detected using the CbCr-I component GMM, after which morphological operations, prior position, and shape information are utilized to segment the hand region, which contains only the detected hand posture. Then the proposed features are extracted from the hand regions, and they are fed into the multiclass SVM classifier as input vectors to build the multiclass SVM classifier model. In the testing stage, the multiclass SVM classifier model is used to classify the hand posture. In addition, SVM is a binary classifier, and it is extended

Fig. 4 Segmented hand region and its corresponding features. **a** Segmented hand region. **b** Skin detection result of segmented hand region. **c** Hand contour. **d** Distances between the centroid and the points on the hand contour

Fig. 5 Hand contours and their features

to recognize m -class postures. We use so called one against one strategy, i.e., $m(m-1)/2$ classifiers are needed to train. Moreover, we choose radial basis function as the kernel function in this paper.

Because the hand sizes are different, the lengths of hand contours are also different, which may lead to a failed recognition. To deal with this problem, the lengths of hand contours need to be resized. L_o and L denote the length of the original hand contour and the length of final hand contour, respectively. If $L_o > L$, the signals are re-sampled to obtain the final length of the hand contour; If $L_o < L$, then linear interpolation is used to resize the hand contour. The process of re-sampling or interpolation S is defined by Eq. (11). If $S(n-1)$ is not an integer, then the final hand contour will be computed by Eq. 12, and if $S(n-1)$ is an integer, the obtained interpolation will be $l_r^n = l_o^{S(n-1)+1}$.

$$S = \frac{L_o - 1}{L - 1} \quad (11)$$

$$l_r^n = \frac{[S(n-1)+1] - (S(n-1)+1)}{[S(n-1)+1] - \lfloor S(n-1)+1 \rfloor} l_o^{[S(n-1)+1]} + \frac{(S(n-1)+1) - \lfloor S(n-1)+1 \rfloor}{[S(n-1)+1] - \lfloor S(n-1)+1 \rfloor} l_o^{\lfloor S(n-1)+1 \rfloor}, \quad (12)$$

where l_r^n is the n th point on the result curve; l_o^n is the n th point on the original curve; $\lceil * \rceil$ presents the ceil of the fractional number $*$, and $\lfloor * \rfloor$ is defined as the floor of the fractional number $*$.

Experimental results

A dataset was constructed to evaluate the performance of the proposed method of hand posture recognition. The dataset consists of 1200 images with ten classes of postures. Three scenes are presented in our dataset, which include the conference room, laboratory, and outdoor. Four-hundred hand postures are captured in each scene. There are ten types of

Fig. 6 Several sample images of our data set

Fig. 7 Skin region detection and hand region segmentation in conference room. **a, d** Original images. **b, e** Results of skin region detection based on the CbCr-I component GMM. **c, f** Results of hand region segmentation

Fig. 8 Skin region detection and hand region segmentation in the lab. **a, d** Original images. **b, e** Results of skin region detection based on the CbCr-I component GMM. **c, f** Results of hand region segmentation

Fig. 9 Skin region detection and hand region segmentation outdoors. **a, d** Original images. **b, e** Results of skin region detection based on the CbCr-I component GMM. **c, f** Results of hand region segmentation

Fig. 10 Hand region segmentation of a hand posture image with user wearing short sleeves. **a** Original images. **b** Segmentation results of gesturing arm. **c** Center line of the gesturing arm. **d** Center line of the

forearm. **e** Horizontal distance curve. **f** Gradient of the horizontal distance curve. **g** Approximate location of the wrist. **h** Results of hand region segmentation

Fig. 11 Skin region detection and hand region segmentation of hand posture images with user wearing short sleeves. **a, d** Original images. **b, e** Segmentation results of gesturing arm. **c, f** Results of hand region segmentation

Table 2 Confusion matrix for ten hand postures using our dataset in the conference room

Hand postures	Prediction										Accuracy (%)
	1	2	3	4	5	6	7	8	9	10	
Test											
1	38	0	0	0	0	0	0	0	1	1	95.0
2	0	39	0	0	0	0	0	0	0	1	97.5
3	0	0	36	1	0	0	1	1	0	1	90.0
4	0	0	3	35	2	0	0	0	0	0	87.5
5	0	0	1	2	37	0	0	0	0	0	92.5
6	0	0	0	0	0	39	0	0	1	0	97.5
7	0	0	4	1	0	0	35	0	0	0	87.5
8	1	1	0	1	0	0	1	35	0	1	87.5
9	0	0	1	0	0	1	0	1	37	0	92.5
10	0	0	0	0	0	0	0	0	0	40	100.0
Average accuracy (%)											92.75

Table 3 Confusion matrix for ten hand postures using our dataset in the lab

Hand postures	Prediction										Accuracy (%)
	1	2	3	4	5	6	7	8	9	10	
Test											
1	38	2	0	0	0	0	0	0	0	0	95.0
2	1	39	0	0	0	0	0	0	0	0	97.5
3	0	1	30	2	5	1	0	0	1	0	75.0
4	0	0	1	37	2	0	0	0	0	0	92.5
5	0	0	0	2	36	0	0	1	1	0	90.0
6	0	0	0	1	1	37	0	0	1	0	92.5
7	0	0	0	0	1	0	37	1	1	0	92.5
8	0	0	1	0	0	1	0	37	1	0	92.5
9	1	0	0	0	0	2	0	0	37	0	92.5
10	1	0	0	0	0	0	0	0	0	39	97.5
Average accuracy (%)											91.75

hand postures in each scene, including both the left and right hand. The number of each class of hand postures of each hand in each scene is 20, and each type of hand posture is different with respect to the position, distance, and rotation. Because we want to utilize hand postures to control the movements of our multifunctional mobile robot, these postures are captured by a laptop camera, which can be used to remotely operate the robot. The size of the image in our data set is 1290×720 pixels. We attempt to design our hand posture recognition system from human centred viewpoint. When the user interacts with the robot through hand postures, the user looks toward the camera and can see the captured image. The user gestures in his/her natural way of making himself/herself comfort-

able. If both face and the gesturing hand are connected in the captured image, even eyes of the user are blocked by the gesturing hand, the user will feel uncomfortable. So, it is assumed that the user's hand is always away from the face in the captured image. When the user is gesturing, they generally turn their palm to the camera, thus it is assumed that the palm of the postures is facing the camera. Some of the images in our dataset are shown in Fig. 6.

Table 4 Confusion matrix for ten hand postures using our dataset outdoors

Hand postures	Prediction										Accuracy (%)
	1	2	3	4	5	6	7	8	9	10	
Test											
1	39	0	0	0	0	0	0	0	0	1	97.5
2	1	37	0	0	0	0	0	1	0	1	92.5
3	0	0	37	1	0	0	0	1	1	0	92.5
4	1	0	1	37	0	0	0	0	0	1	92.5
5	0	0	1	2	36	0	1	0	0	0	90.0
6	0	0	0	1	1	37	0	1	0	0	92.5
7	0	0	0	1	0	0	37	2	0	0	92.5
8	0	0	1	0	1	0	0	37	1	0	92.5
9	0	0	0	0	0	1	0	0	38	1	95.0
10	0	0	0	1	0	0	0	0	1	38	95.0
Average accuracy (%)											93.25

Table 5 Comparisons of our approach with other methods for ten hand postures in the conference room

Types	1	2	3	4	5	6	7	8	9	10
HOG+SVM [44]	97.5	92.5	90.0	97.5	92.5	97.5	85.0	95.0	92.5	95.0
Proposed method	97.5	97.5	77.5	87.5	95.0	97.5	95.0	95.0	95.0	90.0
LeNet-5 [62]	82.5	92.5	72.5	87.5	75.0	92.5	70.0	75.0	85.0	95.0
ResNet-18 [63]	97.5	95.0	95.0	97.5	95.0	97.5	85.0	97.5	95.0	100.0

Hand segmentation

Hand region segmentation of hand postures with long sleeves

After the detection of skin regions using the skin color model mentioned above, the prior knowledge that the position of the face is higher than hand is employed, i.e., the hand being lower than the face. Here, we assume that the gesturing hand is lower than face, as in most of general cases. Based on this assumption, the model automatically obtains the second lower skin area as the detected hand region. In other words, the hand region is segmented as a region of interest (ROI) from the image. The result of hand segmentation is shown in Figs. 7, 8, and 9. Our segmentation algorithm performs well in demonstrating the effectiveness of the proposed method.

Hand region segmentation of hand postures with short sleeves

The proposed hand segmentation algorithm is employed to segment short-sleeved hand postures. Figure 10 shows the

hand region segmentation results of a short-sleeved hand posture image.

The proposed segmentation algorithm was utilized to segment ten types of hand postures with short sleeves, and some hand region segmentation results are shown in Fig. 11. We can see that the proposed segmentation method can segment the hand region from both images of users wearing long and short sleeves, which demonstrate the effectiveness of the proposed segmentation algorithm.

Multiclass hand posture recognition

The presented method was evaluated in our dataset. Our dataset comprises ten types of hand posture images in the conference room scene, in the lab and outdoors. There are 1200 images for the three scenes in total. Ten types of hand postures are presented in each scene, and each type of hand posture consists of both 20 left hand images and 20 right hand images. Furthermore, the 18 left-hand images and 18 right-hand images of each type of hand postures in each scene are used to train the multiclass SVM classifier, while the remaining two images for each side are used to test the trained model.

Table 6 Comparisons of our approach with the other methods for ten hand postures in the lab

Types	1	2	3	4	5	6	7	8	9	10
HOG+SVM [44]	87.5	82.5	95.0	97.5	87.5	100.0	97.5	90.0	90.0	100.0
Proposed method	92.5	92.5	95.0	87.5	77.5	90.0	100.0	95.0	87.5	100.0
LeNet-5 [62]	77.5	67.5	77.5	80.0	82.5	90.0	77.5	70.0	82.5	100.0
ResNet-18 [63]	92.5	100.0	92.5	97.5	97.5	100.0	92.5	97.5	95.0	100.0

Table 7 Comparisons of our approach with the other methods for ten hand postures outdoors

Types	1	2	3	4	5	6	7	8	9	10
HOG+SVM [44]	95.0	97.5	95.0	92.5	90.0	100.0	95.0	97.5	95.0	97.5
Proposed method	95.0	100.0	92.5	85.0	95.0	97.5	97.5	87.5	92.5	90.0
LeNet-5 [62]	92.5	80.0	90.0	72.5	72.5	92.5	87.5	92.5	95.0	87.5
ResNet-18 [63]	100.0	97.5	92.5	92.5	85.0	95.0	87.5	85.0	97.5	97.5

Table 8 Training time of different methods for hand postures in different scenes

Types	Conference room(s)	Laboratory (s)	Outdoor (s)
HOG+SVM [44]	2602.7	2542.3	2446.2
Proposed method	15.4	15.6	15.0
LeNet-5 [62]	16.9	17.0	17.4
ResNet-18 [63]	917.4	892.2	891.7

Hence, 360 images are used for training, and 40 images are used to test the trained classifier in three scenes.

The process of hand posture recognition consists of training and testing stages, as shown in Fig. 1. The training procedure is described as follows. First, the hand region is segmented from the image. Then, the proposed features are extracted from the hand region, and sent as input vectors to the multiclass SVM classifier to build the model. In the testing stage, after the hand region is segmented from the image, the features are extracted from the segmented image that contains the hand posture only. Finally, the features are fed into the multiclass SVM classifier model to recognize the hand posture.

Experiments are performed using tenfold cross validation on a computer with Intel® Xeon(R) Gold 6254 CPU @ 3.10GHz. The performance of the hand posture recognition in the conference room, lab, and outdoors are shown in Tables 2, 3 and 4, respectively. The proposed algorithm has good performance, and the average accuracies of hand posture recognition are 92.75%, 91.75% and 93.25%, respectively, which demonstrates the effectiveness of the presented approach.

The hand postures, like the hand postures in Fig. 6e, g and l, have two stretched fingers, however, they depict different hand postures, and they can be recognized using the proposed method. However, these postures cannot be rec-

ognized by the method that recognizes the hand posture by counting the number of zero-to-one (black-to-white) transitions of the constructed the binary circle with the center at the COG of the hand region, and the radius equal to 70% of the farthest distance in the hand region from the COG [16]. Moreover, some hand postures with the same number of stretched fingers, that denote different hand postures, have been recognized using the proposed hand posture recognition method in the conference room, in the lab, and outdoors, respectively, which demonstrates the effectiveness of the proposed hand posture recognition approach.

To evaluate the performance of the proposed method, some comparison experiments were carried out. The performance comparison is shown in Tables 5, 6, 7 and 8. The average accuracies of the proposed method in the conference room, lab, and outdoors are 92.75%, 91.75% and 93.25%, respectively. Whereas the average accuracies of the HOG and SVM method [44] in the conference room, lab, and outdoors are 93.5%, 92.75%, and 95.5%, respectively. The HOG and SVM [44] and the proposed method have a similar recognition accuracy. The dimension of the proposed feature is smaller than HOG. In contrast with HOG, the proposed feature has a dimension of 1×60 rather than 1×20736 . Thus, the proposed method significantly reduces calculation. The average accuracies of the proposed method in the conference room, lab, and outdoors are 92.75%, 91.75% and 93.25%,

respectively. Whereas the average accuracies of LeNet-5 [62] in the conference room, lab, and outdoors are 82.75%, 80.50% and 86.25%, respectively. In most cases, the accuracies of the proposed method are higher than LeNet-5 [62] method, perhaps because the number of samples is small. The average accuracies of the proposed method in the conference room, lab, and outdoors are 92.75%, 91.75% and 93.25%, respectively. Whereas the average accuracies of ResNet-18 [63] in the conference room, lab, and outdoors are 95.5%, 96.5% and 93.0%, respectively. The proposed method and ResNet-18 [63] method have a similar recognition accuracy. However, the training time of the proposed method is lower than ResNet-18 [63] method.

Conclusions

A novel method of hand posture recognition based on a new hand shape distribution feature and the CbCr-I component GMM skin color detection is presented. To reduce the effect of variable illumination, the CbCr-I component GMM is proposed. Moreover, the hand region is segmented as a region of interest using the CbCr-I component GMM and the adaptive threshold in the segmentation step. Hand regions can be segmented from both images of user wearing long and short sleeves. Subsequently, a new hand shape distribution feature was proposed based on the hand contour described in polar coordinates in the recognition step. This feature only utilized hand contour information to represent hand postures. The proposed feature is robust to the location of the origin, when the origin is located on the palm of the hand. Finally, a multiclass SVM classifier is employed to recognize hand postures. Because the contours of the hand regions are closed curves, the proposed method can deal with the false problem in some shape-based methods. To evaluate the performance of the proposed method, we built a dataset and conducted experiments of hand posture recognition to test the model. The experimental results showed the effectiveness of the proposed method, demonstrating that this algorithm can recognize hand postures, particularly in the cases where the same number of stretched fingers represents different hand postures.

Acknowledgements The work is supported by the National Natural Science Foundation of China (Grant no. 91748201 and Grant no. 51775011) and Natural Science Foundation of Beijing Municipality (Grant no. 3192017).

Declaration

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Pavlovic VI, Sharma R, Huang TS (1997) Visual interpretation of hand gestures for human–computer interaction: a review. *IEEE Trans Pattern Anal Mach Intell* 19(7):677–695
2. Hasan H, Abdul-Kareem S (2014) Retracted article: human–computer interaction using vision-based hand gesture recognition systems: a survey. *Neural Comput Appl* 25(2):251–261
3. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. *Artif Intell Rev* 43(1):1–54
4. Cheok MJ, Omar Z, Jaward MH (2019) A review of hand gesture and sign language recognition techniques. *Int J Mach Learn Cybern* 10(1):131–153
5. Parvathy P, Subramaniam K, Venkatesan GKDP, Karthikaikumar P, Varghese J, Jayasankar T (2020) Development of hand gesture recognition system using machine learning. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-020-02314-2>
6. Pisharady PK, Saerbeck M (2015) Recent methods and databases in vision-based hand gesture recognition: a review. *Comput Vis Image Underst* 141:152–165
7. Agah A (2000) Human interactions with intelligent systems: research taxonomy. *Comput Electr Eng* 27(1):71–107
8. Shang W, Cao X, Ma H, Zang H, Wei P (2016) Kinect-based vision system of mine rescue robot for low illuminous environment. *J Sens* 2016:1–9
9. Chen X, Koskela M (2013) Online rgb-d gesture recognition with extreme learning machines. In: *ICMI '13: 2013 international conference on multimodal interaction*, Sydney, Australia. SIGCHI, Association for Computing Machinery, New York, NY, United States, pp 467–474
10. Palacios JM, Sagüés C, Montijano E, Llorente S (2013) Human–computer interaction based on hand gestures using rgb-d sensors. *Sensors* 13(9):11842–11860
11. Trigueiros P, Ribeiro F, Reis LP (2015) Generic system for human–computer gesture interaction: applications on sign language recognition and robotic soccer refereeing. *J Intell Robot Syst* 80(3–4):573–594
12. Badi HS, Hussein S (2014) Hand posture and gesture recognition technology. *Neural Comput Appl* 25(3–4):871–878
13. Chang C-C, Chen J-J, Tai W-K, Han C-C et al (2006) New approach for static gesture recognition. *J Inf Sci Eng* 22(5):1047–1057
14. Girondel V, Bonnaud L, Caplier A (2006) A human body analysis system. *EURASIP J Adv Signal Process* 2006(1):061927
15. Manigandan Mr, Jackin I Manju (2010) Wireless vision based mobile robot control using hand gesture recognition through perceptual color space. In: *2010 International Conference on Advances in Computer Engineering*. IEEE, pp 95–99
16. Malima AK, Özgür E, Çetin M (2006) A fast algorithm for vision-based hand gesture recognition for robot control. In: *2006 IEEE*

- 14th signal processing and communications applications, Antalya, Turkey, 2006, IEEE, pp 1–4
17. Ghidary SS, Nakata Y, Saito H, Hattori M, Takamori T (2002) Multi-modal interaction of human and home robot in the context of room map generation. *Auton Robots* 13(2):169–184
18. Ghidary SS, Nakata Y, Takamori T, Hattori M (2000) Human detection and localization at indoor environment by home robot. In: SMC 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no. 0, vol 2). IEEE, pp 1360–1365
19. Thangali A, Sclaroff S (2009) An alignment based similarity measure for hand detection in cluttered sign language video. In: 2009 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 89–96
20. Wu Y, Huang TS (2000) View-independent recognition of hand postures. In: Proceedings IEEE conference on computer vision and pattern recognition. CVPR 2000 (Cat. No. PR00662), vol 2. IEEE, pp 88–94
21. Myung-Ho J, Kang H-B (2012) Emotional interaction with a robot using facial expressions, face pose and hand gestures. *Int J Adv Robot Syst* 9(3):95
22. Qi J, Xu K, Ding X (2017) Hand detection from cluttered images based on a hierarchical strategy. In: Chinese intelligent automation conference. Springer, pp 783–791
23. Adithya V, Rajesh R (2018) An efficient method for hand posture recognition using spatial histogram coding of nct coefficients. In: 2018 IEEE recent advances in intelligent computational systems (RAICS). IEEE, pp 16–20
24. Zivkovic Z (2004) Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004., vol 2. IEEE, pp 28–31
25. Chen F-S, Chih-Ming F, Huang C-L (2003) Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image Vis Comput* 21(8):745–758
26. Cutler R, Turk M (1998) View-based interpretation of real-time optical flow for gesture recognition. In: Proceedings third IEEE international conference on automatic face and gesture recognition. IEEE, pp 416–421
27. Wang K, Lei X, Fang Y, Li J (2013) One-against-all frame differences based hand detection for human and mobile interaction. *Neurocomputing* 120:185–191
28. Rahmat RW, Al-Tairi ZH, Saripan MI, Sulaiman PS (2012) Removing shadow for hand segmentation based on background subtraction. In: 2012 international conference on advanced computer science applications and technologies (ACSAT). IEEE, pp 481–485
29. Haines TSF, Xiang T (2013) Background subtraction with dirichletprocess mixture models. *IEEE Trans Pattern Anal Mach Intell* 36(4):670–683
30. Singla N (2014) Motion detection based on frame difference method. *Int J Inf Comput Technol* 4(15):1559–1565
31. Nickel K, Stiefelbogen R (2007) Visual recognition of pointing gestures for human–robot interaction. *Image Vis Comput* 25(12):1875–1884
32. Corradini A (2001) Dynamic time warping for off-line recognition of a small gesture vocabulary. In: Proceedings IEEE ICCV workshop on recognition, analysis, and tracking of faces and gestures in real-time systems. IEEE, pp 82–89
33. Silanon K (2017) Thai finger-spelling recognition using a cascaded classifier based on histogram of orientation gradient features. *Comput Intell Neurosci* 2017:1–11
34. Lahiani H, Kherallah M, Neji M (2017) Hand gesture recognition system based on local binary pattern approach for mobile devices. In: International conference on intelligent systems design and applications. Springer, pp 180–190
35. Kuizhi Mei LX, Li B, Lin B, Wang F (2015) A real-time hand detection system based on multi-feature. *Neurocomputing* 158:184–193
36. Haile H-BU, Victor A-R (2016) Real-time hand posture recognition for human–robot interaction tasks. *Sensors* 16(1):36
37. Hernandez-Belmonte UH, Ayala-Ramirez V (2016) Feature selection using genetic algorithms for hand posture recognition. In: Mexican conference on pattern recognition. Springer, pp 208–218
38. Ding Y, Pang H, Xuechun W (2011) Static hand-gesture recognition using hog and improved lbp features. *Int J Digit Content Technol Appl* 5(11):236–243
39. Wang C-C, Wang K-C (2007) Hand posture recognition using adaboost with sift for human robot interaction. In: Recent progress in robotics: viable robotic service to human. Springer, pp 317–329
40. Vishwakarma DK (2017) A hybrid approach for the recognition of hand postures using texture and skin saliency map. In: 2017 international conference on intelligent sustainable systems (ICISS). IEEE, pp 434–437
41. Reddy DA, Sahoo JP, Ari S (2018) Hand gesture recognition using local histogram feature descriptor. In: 2018 2nd international conference on trends in electronics and informatics (ICOEI). IEEE, pp 199–203
42. Cao J, Siquan Yu, Liu H, Li P (2016) Hand posture recognition based on heterogeneous features fusion of multiple kernels learning. *Multimed Tools Appl* 75(19):11909–11928
43. Vishwakarma DK, Singh K et al (2016) A framework for recognition of hand gesture in static postures. In: 2016 international conference on computing, communication and automation (ICCCA). IEEE, pp 294–298
44. Feng K, Yuan F (2013) Static hand gesture recognition based on hog characters and support vector machines. In: 2013 2nd international symposium on instrumentation and measurement, sensor network and automation (IMSNA). IEEE, pp 936–938
45. Tiantian L, Jinyuan S, Runjie L, Yingying G (2015) Hand gesture recognition based on improved histograms of oriented gradients. In: The 27th Chinese control and decision conference (2015 CCDC). IEEE, pp 4211–4215
46. Ahuja MK, Singh A (2015) Static vision based hand gesture recognition using principal component analysis. In: 2015 IEEE 3rd international conference on MOOCs, innovation and technology in education (MITE). IEEE, pp 402–406
47. Chen S, Ma H, Han C (2017) A real-time hand postures estimation method. In: 2017 IEEE 7th annual international conference on CYBER technology in automation, control, and intelligent systems (CYBER). IEEE, pp 1059–1064
48. Zhao H, Hu J, Zhang Y, Cheng H (2017) Hand gesture based control strategy for mobile robots. In: 2017 29th Chinese control and decision conference (CCDC). IEEE, pp 5868–5872
49. Yin X, Xie M (2007) Finger identification and hand posture recognition for human–robot interaction. *Image Vis Comput* 25(8):1291–1300
50. Zhou Y, Jiang G, Lin Y (2016) A novel finger and hand pose estimation technique for real-time hand gesture recognition. *Pattern Recognit* 49:102–114
51. Zhou Y, Jiang G, Xu G, Lin Y (2014) Hand gesture recognition based on the parallel edge finger feature and angular projection. In: Asian conference on computer vision. Springer, pp 206–217
52. Ozturk O, Aksac A, Ozyer T, Alhajj R (2015) Boosting real-time recognition of hand posture and gesture for virtual mouse operations with segmentation. *Appl Intell* 43(4):786–801
53. Liu Q, Xu D, Li Z, Zhou P, Zhou J, Xu Y (2015) A new distance metric learning algorithm for hand posture recognition. In: 2015 3rd international conference on mechatronics and industrial informatics (ICMII 2015). Atlantis Press

54. Wang X, Yan K (2019) Immersive human–computer interactive virtual environment using large-scale display system. *Future Gener Comput Syst* 96:649–659
55. Liu Y, Wang X, Yan K (2018) Hand gesture recognition based on concentric circular scan lines and weighted k-nearest neighbor algorithm. *Multimed Tools Appl* 77(1):209–223
56. Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. *Pattern Recognit* 40(3):1106–1122
57. Hsu R-L, Abdel-Mottaleb M, Jain AK (2002) Face detection in color images. *IEEE Trans Pattern Anal Mach Intell* 24(5):696–706
58. Dai Y, Nakano Y (1996) Face-texture model based on sgld and its application in face detection in a color scene. *Pattern Recognit* 29(6):1007–1017
59. Salazar-Colores S, Ramos-Arreguín J-M, Pedraza-Ortega J-C, Rodríguez-Reséndiz J (2019) Efficient single image dehazing by modifying the dark channel prior. *EURASIP J Image Video Process* 2019(1):66
60. Rekha J, Bhattacharya J, Majumder S (2011) Hand gesture recognition for sign language: a new hybrid approach. In: *International conference on image processing, computer vision, & pattern recognition (IPCVR 2011)*, 18–21 July, Las Vegas, Nevada, United States of America, pp 80–86
61. Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 6:679–698
62. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
63. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.