Hand gesture recognition algorithm combining hand-type adaptive algorithm and effective-area ratio for efficient edge computing

Qiang Zhang,a Shanlin Xiao,b Zhiyi Yu,a,b,* Huanliang Zheng,b and Peng Wangc

aSun Yat-sen University, School of Microelectronics Science and Technology, Zhuhai, China
bSun Yat-sen University, School of Electronics and Information Technology, Guangzhou, China
cHarbin University of Science and Technology, School of Electrical and Electronic Engineering, Harbin, China

Abstract. Most existing gesture recognition algorithms have low recognition rates under rotation, translation, and scaling of hand images as well as different hand types. We propose a new hand gesture recognition algorithm that combines the hand-type adaptive algorithm and effective-area ratio based on feature matching. Samples are divided into several groups according to the subjects’ palm shapes and the algorithm is trained using self-collected data. The user’s hand type is paired with one of the sample libraries by the hand-type adaptive algorithm. To further improve the accuracy, the effective-area ratio of the gesture is calculated based on the minimum bounding rectangle, and the preliminary gesture is recognized by the effective-area ratio feature method. The results of experiments demonstrate that the proposed algorithm could accurately recognize gestures in real time and exhibits good adaptability to different hand types. The overall recognition rate is over 94%. The recognition rate still exceeds 93% when hand gesture images are rotated, translated, or scaled. © 2021 SPIE and IS&T [DOI: 10.1117/1.JEI.30.6.063026]

Keywords: gesture recognition; effective-area ratio; hand-type adaptive; area-perimeter ratio; edge computing.

Paper 210225 received Jun. 30, 2021; accepted for publication Nov. 23, 2021; published online Dec. 23, 2021.

1 Introduction

Human–computer interaction is currently realized mainly through a mouse, keyboard, remote control, and touch screen. However, actual interpersonal communication is primarily performed through a more natural and intuitive noncontact manner, such as sound and physical movements, which are considered to be flexible and efficient. Researchers have been attempting to develop machines that recognize human intentions through noncontact communication modes as humans do, such as by sound,1 facial expressions,2 body language,3 and gestures.4,5 Among these modes, hand gestures6 are an important part of human language, and hence, the development of hand gesture recognition affects the nature and flexibility of human–computer interaction.7–11

In the past few decades, the gesture recognition method is typically based on the angle and position information obtained through data gloves.12 However, this method is expensive and uses wearable sensors, making the method inconvenient. Hand gesture data are also collected using optics camera13–15 or radar.16–19 Optical-based gesture recognition mainly uses cameras to capture gesture images and then applies machine-learning methods20,21 for feature extraction and recognition. Coelho et al.14 used Kinect to capture RGB and depth images of hand gestures. By contrast, machine-vision-based noncontact recognition methods are currently popular as they have the advantages of low cost, convenience, and comfort for the human body.

In this paper, we propose a hand-adaptive algorithm that can significantly reduce the impact of hand type in gesture recognition. It has advantages when the number of recognitions is small.

*Address all correspondence to Zhiyi Yu. yuzhiyi@mail.sysu.edu.cn
Compared with the current popular artificial intelligence and deep learning algorithms, this algorithm has lower hardware requirements and is more suitable for embedded edge computing, which is also very popular at present.

The contributions of this paper are listed as follows:

1. We have designed a hand-adaptive algorithm to solve the problem of low recognition rate caused by different hand types that traditional simple algorithms cannot cope with. Therefore, the proposed algorithm does not directly recognize the input gesture images, but first classifies them by hand type, and then uses different sample libraries for recognition according to different hand types. This improves the overall recognition rate with almost negligible resource consumption.

2. Using gesture prerecognition can reduce the amount of calculation and hardware resources. The proposed algorithm uses the shortcut feature to select three candidate gestures and then uses a high-precision and complex algorithm to determine the final gesture among the candidate gestures. It not only improves the speed of gesture recognition but also ensures the accuracy of recognition.

3. We selected area–perimeter ratio and effective-area ratio to obtain features with low computational complexity and the accuracy loss is almost negligible. These features are not sensitive to most transformations, so they can deal with the rotation, translation, and scaling of the image during the gesture recognition process. Since the amount of calculation to obtain them is small, the recognition speed can be higher.

4. The proposed algorithm is implemented based on resource-constrained hardware platforms. In the comparison experiment, FPGA is used to implement the proposed algorithm and compared with the Intel Core series chip hardware platform, which has a lot of advantages in hardware resources. FPGA achieved a recognition rate of 94.99%, comparable to Intel. Moreover, the time for gesture recognition is significantly reduced, and the time cost is less than half of Intel’s under such a huge hardware resource gap.

2 Relevant Work

Extraction of gesture features is one of the most important aspects of gesture recognition. In general, the accuracy and range of gesture recognition depend on the amount of gesture feature information extracted. Many algorithms have been proposed to recognize hand gestures from images. There are also many simple image processing algorithms such as the proposed algorithm. They also tried to find a simple and practical algorithm such as convex hull.22,23 Woun Bo Shen and Tan Guat Yew used a convex hull in the feature extraction stage. This algorithm is simple and efficient, but the number of gestures recognized is small. In addition, based on Kinect’s method of detecting the angle of a hull,24 and this algorithm also has the disadvantage of a low number of recognition gestures. The hardware equipment required for the above algorithms is expensive and therefore not conducive to popularization.

Furthermore, these algorithms are not sufficiently intuitive to represent the hand gestures formed by different hand types. Other algorithms consider fingers as the features and they are detected on the basis of ridge detection,25–30 a circle drawn on the hand centroid,31,32 or convex decomposition.33 However, the method in Ref. 34 is time-consuming, while the others28–32 are not effective to handle fingers with distortion. Subsequent classification algorithms28-32 are learning-based, which require many training images for each class. Moreover, algorithms that use rule classifiers and zero training images29–31 lack adaptability for certain gestures with distortion and varying postures. Therefore, a balance should be maintained between convenience and robustness. Zhang et al.34 proposed a recognition algorithm based on Hu moment invariants for rotating images. The paper improves the algorithm by changing the characteristic value of Hu algorithm and calculating the similarity between the image to be recognized and the template image. However, this method is not intuitive enough and highly accurate and real-time detection performance cannot be guaranteed using the Hu moment feature alone. Dardas and Georganas35 performed scale-invariant feature transformation (SIFT) and vectorization feature extraction on images then used feature packets and multiclass support vector machines to recognize gestures.
The SIFT algorithm has a higher recognition rate, but the computational complexity of the algorithm is higher, so the recognition speed is lower and the real-time performance is poor.

To recognize and classify signatures of hand gestures, numerous techniques have been applied such as machine learning, principal component analysis, and differentiate/cross-multiply algorithms. Conventional supervised machine learning extracts and classifies gestures using predefined characteristic parameters (features). However, the optimal features are unknown in many cases, resulting in a significant variation of the performance of the classifier depending on the selected features. Some deep learning algorithms are large in scale and require high hardware performance and require a large number of training samples. Some deep networks require both training and GPU support for online deployment, which pose high hardware demand and thus is not conducive to small, embedded artificial intelligence systems. The above algorithms did not design and select features to reduce the amount of computation. They also failed to solve the problem that complex gesture recognition algorithms cannot be used or lack real-time performance when applied to embedded artificial intelligence systems with limited hardware resources.

3 Proposed Algorithm

In this study, gesture recognition is divided into two parts: (1) establishment of a sample library by the process shown in Fig. 1. Three sample libraries are built according to the hand-type classification. (2) Gesture recognition by the process shown in Fig. 2. The hand-adaptive algorithm matches the suitable sample library for the user’s hand type, thus reducing the interference of hand type on gesture recognition accuracy. The number of samples that are finally identified is reduced by preliminary recognition, thereby allowing a fast recognition process.

3.1 Building Libraries for Hand-Type Adaptation

First, subjects are selected and their palms are measured. Next, the subjects are divided into three groups according to the palm size of the palm, namely, slim, normal, and broad. Then, the gesture features of the three groups of subjects are calculated separately to establish the sample library.

3.1.1 Selection of subjects

In this study, 40 subjects were selected after obtaining informed consent from them: 27 young people (13 females and 14 males) aged 15 to 35, 8 middle-aged people (4 females and 4 males), aged 36 to 55, and 5 elderly (3 males and 2 females) aged 56 to 70. The collected samples are presented in Fig. 3.
3.1.2 Obtaining hand-type data

First, the maximum length of the palm, \(L_1\), is measured from the longest fingertip to the root of the palm, which is the first distinct line between the palm and the wrist, close to the palm. Then, the maximum length of the finger, \(L_2\), is measured from the longest fingertip to its finger root. Finally, the palm width \(L_3\) is measured. The measurement diagram is shown in Fig. 4. To reduce the error, an average of three measurements is taken.

The \(L_1\), \(L_2\), and \(L_3\) measurements provide a peripheral contour convex hull of the entire hand. In image processing, a convex hull can be considered a convex set that surrounds the outermost layer of the image. The measurement of the peripheral contour convex hull of the
hand is shown in Fig. 5. The convex hull defect and its starting point are determined. The relative positions of the palm and fingers are determined and the center point and contour of the palm are calibrated. The center point and the radius of the palm are used to obtain the coordinates of the lowest point of the palm contour, following which the image of the wrist part below the lowest point is eliminated. The ordinates of the middle finger fingertip $A_1$, the palm contour lowest point $A_2$, the middle finger convex hull defect $A_3$, and the palm center point $A_0$ are obtained. $L_1$, $L_2$, and $L_3$ are calculated as follows:

$$
\begin{align*}
L_1 &= A_1 - A_3 \\
L_2 &= A_1 - A_2 \\
L_3 &= 2 \times (A_0 - A_3)
\end{align*}
$$

(1)

### 3.1.3 Classification of hand type

On the basis of a large number of sample statistics, the $L_2$ to $L_1$ ratio is multiplied by 0.3 and the $L_3$ to $L_1$ ratio is multiplied by 0.7 to obtain different hand types that can more accurately reflect the human hand types. The multiplication factors are empirical values. The subjects are divided into the following three groups by weighting calculation using Eq. (2): slim, normal, or broad. Table 1 lists the measurement and grouping of the selected 40 subjects:
3.1.4 Building three sample libraries

A total of 360 images were used to build the sample library, and there was no angle change. The algorithm designed in this paper extracts nine dimensional features, including area-perimeter ratio $C$, effective-area ratio $E$, and the seventh-order Hu moment $Hu_1$, $Hu_2$, $Hu_3$, ..., $Hu_7$. Calculate the feature values for the nine hand gestures in each group and take the midvalues as feature vector $O = [C, E, Hu_1, Hu_2, ..., Hu_7]$. The feature vectors of the slim group, normal group, and broad group are denoted by $O_S$, $O_N$, and $O_C$, respectively. The mid-value is obtained according to Eq. (3), where $X$ denotes the mid-value, $x$ denotes the gesture sequence number, and $n$ denotes the total number subjects in the group:

$$X = \begin{cases} 
x_{\frac{x+n}{2}}, & n \in \text{Odd} \\
x_{\frac{x+n}{2}+1}\frac{n}{2}, & n \in \text{Even}
\end{cases}$$

(3)

3.2 Gesture Recognition

In this study, the gesture image of only the palm and no other part, such as the arm, is processed. The gesture image is median filtered in the image preprocessing stage, and then the image is converted from the RGB color space to the YCbCr color space, because the skin color has good clustering in the YCbCr color space, which enables segmentation of the gesture by the threshold. The result is presented in Fig. 6, and its distribution satisfies Eq. (4). Then, the morphological operation is performed on the segmented image to ensure regularization of the gesture pixels to

**Table 1** Hand parameters and grouping.

<table>
<thead>
<tr>
<th>Hand type</th>
<th>Maximum length of palm (cm)</th>
<th>Maximum length of finger (cm)</th>
<th>Palm width (cm)</th>
<th>Number of subjects (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slim</td>
<td>16.3 to 18.6</td>
<td>7.5 to 8.3</td>
<td>7.2 to 8.0</td>
<td>9</td>
</tr>
<tr>
<td>Normal</td>
<td>16.8 to 19.0</td>
<td>7.7 to 8.5</td>
<td>6.7 to 8.0</td>
<td>24</td>
</tr>
<tr>
<td>Broad</td>
<td>18.2 to 19.6</td>
<td>8.1 to 9.3</td>
<td>7.6 to 8.3</td>
<td>7</td>
</tr>
</tbody>
</table>

$$\text{Type} = (0.3L_2 + 0.7L_3) \frac{1}{L_1}. \quad (2)$$
meet the accuracy requirements of the subsequent operations. Through this series of operations, the feasibility of gesture recognition is ensured:

\[
50 \leq Y \leq 255, \quad 87 \leq Cb \leq 142, \quad 132 \leq Cr \leq 151. \tag{4}
\]

### 3.2.1 Feature extraction

In previous studies,\textsuperscript{28-30} the Euclidean distance between a pixel and its nearest boundary in linear time was used to extract gesture features. Here, we propose other features as follows.

**Area–perimeter ratio.** The area–perimeter ratio \( C = \frac{S}{L} \) is not sensitive to the scaling and rotation of gestures, and it can discriminate between hand types well.

Perimeter \( L \) is calculated as follows:

\[
L = \sum\sum f(x, y), f(x, y) = \begin{cases} 1, & (x, y) \in V \\ 0, & (x, y) \notin V \end{cases}
\tag{5}
\]

Area \( S \) is calculated as follows:

\[
S = \sum\sum q(x, y), q(x, y) = \begin{cases} 1, & (x, y) \in R \\ 0, & (x, y) \notin R \end{cases}
\tag{6}
\]

In Eqs. (5) and (6), \( V \) represents the pixel area of the gesture edge, indicated in blue in Fig. 7. \( R \) represents the gesture pixel area, indicated in white in Fig. 7. Hence, the first important parameter of this study, the area–perimeter ratio, is obtained. Noise and light factors adversely affect gesture segmentation, producing burrs at the edge of the gesture. However, this effect is negligible.

**Effective–area ratio.** Unlike Erdem Yavuz\textsuperscript{26} and Jiajun Zhang,\textsuperscript{27} we use the effective-area ratio of the gesture as a feature. The effective-area ratio of the gesture is defined as the ratio of the gesture area to the area of the minimum bounding rectangle (MBR, which is the rectangle that can contain the entire gesture):

\[
E = \frac{S_{\text{hand}}}{S_{\text{MBR}}}
\tag{7}
\]

In the above equation, \( E \) represents the effective-area ratio at acceptable noise levels. \( S_{\text{hand}} \) is the sum of white pixels in the entire gesture area, and \( S_{\text{MBR}} \) is the sum of all the pixels in the MBR.

![Diagram of gesture feature](https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging)
in the binary image, as shown in Fig. 7. The reason for introducing the concept of the effective-area ratio is to achieve high accuracy and control the noise within an acceptable range.

The Hu invariant moment algorithm is effective for image recognition. It describes the picture from the overall feature, and the seventh-order Hu invariant moments remain unchanged for the rotation, translation, and scale transformation of the image. Therefore, the Hu invariant moment algorithm extracts mathematical features that are constant for both image rotation and scaling. This method has the advantages of good stability and accurate recognition in the gesture recognition process and is suitable for discriminating gestures with small variation.

The \((p + q)\)-order geometric distance of a digital image \(f(x, y)\) is defined as follows:

\[
m_{pq} = \sum_x \sum_y x^p y^q f(x, y),
\]

where \(p, q = 0,1,2\).

The geometric center distance is

\[
\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y).
\]

The centroid is \((\bar{x}, \bar{y})\)

\[
\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}.
\]

In Eq. (10), \(m_{10}, m_{01}\) are the first-order geometric moments of the image, and \(m_{00}\) is the zeroth-order geometric distance of the image. For binary images, the geometric center of the image is point \((\bar{x}, \bar{y})\). \(m_{pq}\) changes with the change of the image. Although \(\mu_{pq}\) has translation invariance, it is sensitive when the image is rotated. Therefore, if the feature is represented directly by the geometric center distance and the normal moment, they cannot make feature parameters have both translation invariance and scaling and rotation invariance. The center moment can be normalized, which is invariant to image rotation, translation, and scaling.

The normalized center moment is

\[
\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}},
\]

where \(p, q = 0,1,2\ldots, r = \frac{p+q+2}{2}\).

The seventh-order invariant moments are defined by the normalized second- and third-order center moments, which are invariant to the transformation, rotation, and scaling of the target. The calculation of the invariant moment of a binary image or gray image is quite complex, which limits its use. To achieve faster invariant moment calculation, in this study, Hu moment extraction on the contour of the gesture is performed.

### 3.2.2 Hand-type adaptation

The classifier used in the following gesture recognition process is mainly template matching. Template matching is mainly realized through the calculation of distance. The hand-type adaptive algorithm is implemented using the area–perimeter ratio of the gesture. To achieve hand-type adaptation, the user needs to input 1-9 gestures in sequence. Then, the area–perimeter ratios of the nine gestures are calculated, which are then used to construct the area–perimeter ratio vector \(C = (c_1, \ldots, c_9)\). This algorithm calculates the Euclidean distance between \(C\) and \(O_S, O_N,\) and \(O_C\), and selects the sample library with the smallest Euclidean distance as the paired sample library. The sample library \(O_S, O_N,\) or \(O_C\) contains nine vectors, and \(C\) is compared with the first element of the nine vectors. Because the first element of the vector is also the area–perimeter ratio, this distance is used to evaluate which sample library is the paired sample library. The Euclidean distance is calculated as follows:
\[ D(C, O_3) = \sqrt{(c_1 - o_{11})^2 + \cdots + (c_{9} - o_{91})^2}, \quad c \in C, \quad o \in O, \quad O \in O_3. \]  

(12)

Equation (12) is chosen as the distance paradigm of \( \|C - O\| \). Similarly, \( D(C, O_N) \) and \( D(C, O_C) \) can be calculated.

### 3.2.3 Gesture preliminary recognition

The main purpose of this step is to reduce the amount of calculation for final recognition, especially when the number of recognized samples is very large. Candidate samples can be quickly determined by the effective-area ratio, thereby greatly reducing the amount of calculation required in the template matching process based on Hu invariant moments and improving the speed of gesture recognition. Through experience, the effective-area ratio can be easily calculated, so gestures can be recognized quickly with a high recognition rate.

Preliminary recognition of gestures: according to Eq. (12), the Euclidean distance of the \( E \) between the current gesture and each gesture in the sample library is calculated. Nine distance values based on \( E \) were obtained. The nine Euclidean distances are sorted from small to large \( \{H_{E1}, \cdots, H_{E9}\} \), and the first three are taken as candidate samples.

### 3.2.4 Gesture final recognition

The final gesture recognition step is as follows: the feature used in this step is the seventh-order Hu moment. The seventh-dimensional feature value in the sample database can be regarded as a middle point. Now the algorithm only needs to operate on the three candidate samples that have been selected. The Euclidean distance \( H_{GV_z} \) of the seventh-order Hu moment between the gesture to be recognized and the candidate sample. It is calculated by Eq. (12). \( H_{GV_z} \) is calculated using Eq. (13), and gesture \( V_z \) in the three candidate samples is the final recognition result:

\[ H_{GV_z} = \min\{H_{GV_1}, H_{GV_2}, H_{GV_3}\}, \quad z = 1, \cdots, 3. \]  

(13)

Figure 8 shows a screenshot of the process of the algorithm recognizing gestures in a video stream.

### 4 Experiment and Results

In this study, nine gestures commonly used in life were selected for recognition experiments, as shown in Fig. 9. In this experiment, 40 subjects were reselected according to the above selection rules. The experiment was conducted under the conditions of stable illumination, less noise, and no face appears in the picture. Before the start of the experiment, 40 male and female subjects

---

**Fig. 8** Algorithm recognition gesture simulation screenshot.
with different palm shapes were selected to establish corresponding gesture sample libraries for each gesture at a distance of 55 cm from a camera, as shown in Fig. 3. For each gesture, three groups A, B, and C sequentially store the area–perimeter ratio, the effective-area ratio, and the seventh-order Hu moments into the gesture sample library. The experimental environment of this study was as follows: Windows 10 system, Intel® Core™ i7-10700F @2.90 GHz hardware platform, with 16 GB RAM. Using an ordinary USB camera as a gesture acquisition device, the whole experiment was implemented on the MATLAB 2012a software platform, TensorFlow, and FPGA, including image preprocessing, hand-type adaptive, and gesture recognition. A Kinect camera was used in Ref. 14, but it is complicated and difficult to commercialize on a large scale due to its high cost.

4.1 Fixed Position

In this paper, the new 40 subjects different from previous 40 subjects at the time of building the sample-library were selected. Each subject made the gestures shown in Fig. 9 at a fixed position (the positive direction of the $y$ axis in Fig. 10, that is, the 0-deg position, was 55 cm from the camera), with 10 experiments conducted per subject. Table 2 presents the recognition rates.

4.2 Rotation Condition

For each subject, five experiments for each gesture (600 experiments for each gesture) are performed at three rotation angles of $-45$ deg, 45 deg, and 90 deg (i.e., the angle between the gesture and the $y$ axis, as shown in Fig. 10). Table 3 presents the recognition rates. Both
Dardas\textsuperscript{35} and Sykora\textsuperscript{36} can identify rotated images, but they are complex and have poor real-time performance. Hu moments are invariant to rotation and scale, so the results are expected to not change with rotation and scale change of the hand. The algorithm in this paper is a combination of multiple features, not just Hu features, so this experiment is necessary.

### 4.3 Different Distances

For each subject, five experiments were performed for each gesture (600 experiments for each gesture) at three distances (the distance from the camera): 40, 70, and 85 cm. The recognition rates are presented in Table 4.

### 4.4 Algorithm Comparison

To better illustrate the innovation and advantages of this algorithm, the following comparison experiments are designed. Related statistics are shown in Fig. 11. Considering the accuracy and real-time requirements of the proposed algorithm, the number of subtypes is set to three. This paper designs a comparison of hand types divided into three subtypes and no subtypes. As shown in Fig. 11(a), the experiment confirmed that the hand types were divided into three subtypes,
which improved the overall accuracy rate by nearly 3%. The proposed algorithm is compared with the same design concept algorithms. As shown in Fig. 11(b), it was compared with two excellent algorithms with similar design concepts. Under the same experimental environment, the recognition rate of this algorithm is slightly higher than the other two algorithms. But the algorithm in this paper is more suitable for offline scenarios such as embedded artificial intelligence. In addition, due to the introduction of candidate gestures, the number of gestures can be expanded while ensuring real-time and accuracy. The response time of the three algorithms in the experiment is basically the same. The proposed algorithm uses candidate gestures to overcome the inherent weaknesses of template matching. For example, the consumption of computing resources increases as the number of templates increases, and it is difficult to guarantee the accuracy and real-time performance of the algorithm. The experiment results are shown in Figs. 11(c) and 11(d). The response time of the algorithm with candidate gestures better than that without candidate gestures.

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Distance</th>
<th>Correct gestures</th>
<th>Accuracy (%)</th>
<th>Distance</th>
<th>Correct gestures</th>
<th>Accuracy (%)</th>
<th>Distance</th>
<th>Correct gestures</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>196</td>
<td>98.00</td>
<td>70</td>
<td>197</td>
<td>98.50</td>
<td>85</td>
<td>193</td>
<td>96.50</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>195</td>
<td>97.50</td>
<td>70</td>
<td>194</td>
<td>97.00</td>
<td>85</td>
<td>193</td>
<td>96.50</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>194</td>
<td>97.00</td>
<td>70</td>
<td>194</td>
<td>97.00</td>
<td>85</td>
<td>192</td>
<td>96.00</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>193</td>
<td>96.50</td>
<td>70</td>
<td>195</td>
<td>97.50</td>
<td>85</td>
<td>192</td>
<td>96.00</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>194</td>
<td>97.00</td>
<td>70</td>
<td>193</td>
<td>96.50</td>
<td>85</td>
<td>191</td>
<td>95.50</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>190</td>
<td>95.00</td>
<td>70</td>
<td>190</td>
<td>95.00</td>
<td>85</td>
<td>187</td>
<td>93.50</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>188</td>
<td>94.00</td>
<td>70</td>
<td>187</td>
<td>93.50</td>
<td>85</td>
<td>184</td>
<td>92.00</td>
</tr>
<tr>
<td>8</td>
<td>40</td>
<td>191</td>
<td>95.50</td>
<td>70</td>
<td>191</td>
<td>95.00</td>
<td>85</td>
<td>188</td>
<td>94.00</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>197</td>
<td>98.50</td>
<td>70</td>
<td>196</td>
<td>98.00</td>
<td>85</td>
<td>196</td>
<td>98.00</td>
</tr>
</tbody>
</table>
The following experiments were conducted with the deep-learning (CNN)\textsuperscript{47–49} and Hu moment algorithms.\textsuperscript{34} Shen et al.\textsuperscript{47} combined CNN with different methods, such as x-ray, and CNN is used for edge detection, feature extraction, and recognition. The network architecture from Dayal et al.\textsuperscript{49} includes five combined convolutional layers, and each convolutional combined layer is composed of several subconvolutional layers, nonlinear ReLU layers, and pooling layers. The double channel CNN (DC-CNN)\textsuperscript{48} has improved the rate of hand gesture recognition and has enhanced the generalization ability of the CNN. Multiple channels can obtain more abandoned information and make the identification more accurate. So, we mainly used the DC-CNN to conduct comparative experiments. The DC-CNN structure is composed of two relatively independent convolution neural network. Each channel contains the same number of convolutional layers and parameters. After the pooling layer, the double channels are respectively connected to a full connection layer and a full connection map is performed. There are 200 images are used for each gesture. Such a small number of samples is unfair to deep learning. However, this comparative experiment is designed to show the advantage of the small sample demand of the proposed algorithm. This experiment is only to illustrate the weaknesses of deep learning, not to explain that the accuracy of deep learning is not high. First, experiments were performed with a fixed position: 0 deg rotation angle and 55 cm distance from the camera. The recognition rates are plotted in Fig. 12. Further, experiments were conducted keeping the subjects’ rotation angles $-45$ deg, 45 deg, and 0 deg for each gesture and with 55 cm fixed distance; the recognition rates are plotted in Fig. 13. Finally, experiments were performed for each subject at distances of 40, 70, and 85 cm for each gesture. The recognition rates are plotted in Fig. 14.

The reason for the low accuracy of gesture 3 and gesture 7 is the feature adopted by the algorithm in this paper. Although the area–perimeter ratio and the effective-area ratio have the advantage of a small amount of calculation, gesture three is more likely to be recognized as gesture two or gesture four. Gesture 7 is more likely to be recognized as gesture 6. Since the overall recognition rate is still good, we comprehensively evaluated the advantages and disadvantages and finally decided to continue using them.

For a more comprehensive comparison, we also summarized the accuracy, response time, and hardware platform in Table 5. The hardware verification platform of the proposed algorithm is Altera’s EP4CE15F17C8N chip and Intel® Core™ i7-10700F @2.90 GHz hardware platform, with 16 GB RAM. The other two algorithms have also been reproduced on the Intel® Core™ i7-10700F @2.90 GHz hardware platform, with 16 GB RAM. The accuracy rate is the average of the three cases (Fig. 15).

![Fig. 12](https://example.com/fig12.png) **Fig. 12** Comparison of recognition rates at a fixed location.

![Fig. 13](https://example.com/fig13.png) **Fig. 13** Comparison of recognition rates under different rotation angles.
We want to provide a simple and efficient algorithm that is often overlooked but urgently needed in some areas. It is not used for complex human–computer conversations, so its accuracy and generalization ability are quite good. As shown in Table 6, its accuracy is very close to depth learning, but the hardware resources it consumes are very small. As shown in Fig. 16, the FPGA resource consumption is relatively small. Therefore, smaller hardware resources can be selected in practical applications. This also coincides with this paper’s attempt to provide a simple and efficient algorithm to be applied in some high-performance and low-power embedded scenes.

### 4.5 Results of Hand-Type Adaptive

The experiment results for hand-type adaptive partial pairing are presented in Figs. 16 and 17. The hand-type adaptive algorithm needed to process nine gestures, but the pairing result can only
be output when the last gesture was processed. Therefore, only the last image is presented here with the final pairing result.

5 Discussion

The experiment results indicated that 40 subjects participated in the experiments, and a total of 5400 gesture images in different situations were processed and identified. The proposed method, which is a simple hand gesture recognition method combined with the hand-type adaptive algorithm, utilizing the effective-area ratio, can realize real-time gesture recognition well. Under the fixed position, the overall recognition rate was more than 94%. The recognition rate was more than 94% under different distances from the camera, and it exceeded 93% at different rotation angles. Moreover, because of the combination of the feature recognition of the Hu moment algorithm, a high recognition rate could be still maintained for gestures with a small degree of differentiation, such as gesture 6, gesture 7, and gesture 8. Nine gesture types presented in Fig. 9 were identified in the same situation, and the recognition time of each gesture was recorded. The average recognition time of the nine gestures was calculated, which was 355.27 ms for the algorithm based on Hu moments, while it was 41.79 ms for the proposed algorithm. This confirms that the proposed method has the potential to expand the scope of gesture recognition in the future.

6 Conclusions

In this study, a simple hand gesture recognition algorithm that combines the hand-type adaptive algorithm and effective-area ratio has been proposed. The sample library is paired using the

<table>
<thead>
<tr>
<th>Resources</th>
<th>Total</th>
<th>Used</th>
<th>Usage rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical unit</td>
<td>15,408</td>
<td>5819</td>
<td>38</td>
</tr>
<tr>
<td>User I/O interface</td>
<td>166</td>
<td>156</td>
<td>94</td>
</tr>
<tr>
<td>Total storage space</td>
<td>516,096</td>
<td>58,880</td>
<td>11</td>
</tr>
<tr>
<td>Multiplier</td>
<td>112</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Phase locked loop</td>
<td>4</td>
<td>1</td>
<td>25</td>
</tr>
</tbody>
</table>

Fig. 16 Pairing result display diagram of hand-type S.

Fig. 17 Pairing result display diagram of hand-type M.
hand-type adaptive algorithm. The effective-area ratio of the target is extracted to realize the initial recognition of the gesture and improve the speed of gesture recognition. By combining the Hu moment feature judgment, gestures with a small degree of differentiation can be well recognized. Experiments showed that the proposed algorithm has a high recognition rate and good robustness under different hand-to-camera distances, rotation angles, and hand types. In particular, the hand-type adaptive algorithm and the initial recognition of gestures enable improvement of the overall recognition rate and speed. The proposed recognition algorithm is simple and easy to implement. It has strong stability and practicability under the condition of relatively stable lightness of the environment and complex background.

However, the proposed algorithm has some limitations. Although the experiment can cope with a complex background, a relatively stable illumination condition is required for effective recognition. In addition, the number of recognized gestures is relatively small. Future work will focus on solving lighting effects and identifying a larger number of gestures.

Acknowledgments
This work was supported in part by the National Key Research and Development Program of China under Grant Nos. 2017YFA0206200 and 2018YFB2202601; in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61834005 and 61902443. This manuscript was approved by the People’s Government of Nangang District, Harbin. The informed consent of all subjects was waived by the People’s Government of Nangang District, Harbin.

References

Qiang Zhang is a PhD candidate at the School of Microelectronics Science and Technology, Sun Yat-sen University, Zhuhai, China. His research interests include machine vision, image processing, and embedded artificial intelligence.

Shanlin Xiao received his BS degree in communications engineering and his MS degree in communications systems from the University of Electronic Science and Technology of China, Chengdu, China, in 2009 and 2012, respectively. He received his PhD in communications and computer engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2017. He is currently an associate research professor at the School of Electronics and Information Technology in Sun Yat-sen University, Guangzhou, China.

Zhiyi Yu is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China, and also with the School of Microelectronics Science and Technology, Sun Yat-sen University, Zhuhai, China.

Huanliang Zheng is with the School of Electronics and Information Technology, Sun Yat-sen University.

Peng Wang is a professor in the School of Electrical and Electronic Engineering in Harbin University of Science and Technology.