# Static Gesture Recognition Algorithm Based on Upper Triangular Image Texture and Recursive Graph

**Cai Yang**

School of Computer and Information Technology, Nanyang Normal University, Nanyang 473061, Henan, China
E-mail: nyyc@163.com

**Abstract.** A static gesture recognition algorithm is proposed based on a recursive graph of the upper triangular image texture, motivated by the low accuracy and robustness of existing algorithms. Firstly, the fingertip localization method based on contour curvature is used to obtain the palm region and then the gesture contour model is established. Secondly, a recurrence plot of the gesture contour sequence is built, which is constructed using the central point and the starting point coordinates. Finally, the texture recognition algorithm is applied to calculate the normalized distance between the recurrence plots of the gesture. The experimental results show that the proposed algorithm can achieve higher recognition accuracy under varying complex backgrounds and illumination. At the same time, when the gesture is in rotation, translation, or scaling, the algorithm has high robustness with a small amount of computation and high efficiency.

**Keywords:** *contour sequence; image texture; normalized feature vector; static gesture recognition; upper triangular.*

## 1 Introduction

With the rapid development of computer technology, more efficient human-computer interaction has emerged. As an active way of human-computer interaction, gesture recognition has been developed from the era of data gloves with assistive devices to the stage of machine vision recognition [1]. Machine vision recognition is divided into three stages: gesture segmentation, gesture feature extraction and gesture recognition. Gesture segmentation is the first step of the whole process; the segmentation results play a crucial role in the final recognition. Gesture feature extraction is an intermediate step based on the acquisition of a variety of feature vectors [2]. Gesture recognition is the last and most important step of the whole process. It is a classification process.

Static gesture recognition is an important part of human-computer interaction and is currently a hot research topic. Different algorithms have been used for the three separate phases as proposed by several experts. Badi [3] has proposed a

Received February 14th, 2017, 1st Revision May 9th, 2017, 2nd Revision June 20th, 2017 , Accepted for publication August 21st, 2017.
 ISSN: 2337-5779, DOI: 10.5614/j.eng.technol.sci.2017.49.3.6

study to explore the utility of a neural network based approach in hand gesture recognition. The proposed system uses two recognition algorithms to recognize a set of six specific static hand gestures, namely: open, close, cut, paste, maximize, and minimize. Raheja [4] improved the hand gesture recognition method by using the Microsoft Kinect, which can be operated robustly in uncontrolled environments and is insensitive to hand variations and distortions. Inspired by the mechanism through which Mother Nature handles gustatory perception in humans, a new model was introduced for clustering and classification of hand gestures based on human taste controlling strategy using an approach of gesture recognition based on skeletal data by employing a nearest neighbor classifier with dynamic time warping [5]. Since then similar approaches have been widely reported in the literature, proposing some practical improvements that lead to better recognition results [6]. Ribó [7] focused his attention on the self co-articulation problem, which had not been addressed before. Moreover, a set of novel features was added in the feature extraction stage. A gesture recognition method was proposed to use in the early stages that sequentially calculates the distance between the input and the training data. The proposed method outputs the recognition result when one candidate has a stronger likelihood of recognition than the other candidates, so that similar incorrect gestures are not output [8]. An efficient framework has been introduced for solving the problem of static gesture recognition based on data obtained from web cameras and Kinect's depth sensor [9].

These algorithms have solved some problems. However, they often have high environmental requirements and the recognition rate is not high when the gesture is rotation, scaling or translation. In order to overcome these shortcomings and improve the robustness of the gesture recognition, a static gesture recognition algorithm is proposed based on upper triangular image texture and recursive graph (UTTRG-SGR). The experimental results show that the proposed method is more accurate than the algorithm in reference [10] by 6.17% and than the algorithm in reference [11] by 5.02%. Also, the running time is reduced by 38.7% and 47.8%. Tests on another data set showed that the proposed algorithm is stable and robust.

## 2 Obtaining the Structured Gesture Sequence

### 2.1 Fingertip Location Based on Contour Curvature

The curvature is defined as the change rate of the tangential angle of a point on the curve relative to the arc length. When the curvature value is larger, the degree of the curve is greater.

Through observing the outline of the gesture it can be seen that the curvature of the finger tip is the largest. Because the curvature has the characteristics of rotation and translation invariance, the maximum value of the curvature is the fingertip [12].

The contour curvature can be indirectly measured through calculating the cosine of the angle between two vectors. And the cosine value of the angle can be calculated by the dot product of the two vectors. The curvature $K(p_i)$ of point $p_i$ on the contour curve can be calculated by Eq. (1).

$$K(p_i) = \frac{\overrightarrow{p_i p_{i-k}} \cdot \overrightarrow{p_i p_{i+k}}}{\left\| \overrightarrow{p_i p_{i-k}} \right\| \left\| \overrightarrow{p_i p_{i+k}} \right\|} \tag{1}$$

Among them, the variables $\overrightarrow{p_i p_{i-k}}$ and $\overrightarrow{p_i p_{i+k}}$ represent two vectors, the variable $p_{i-k}$ is the first k point before $p_i$, and the variable $p_{i+k}$ is the first k point behind $p_i$.

Curvature analysis requires a high profile and there may be some problems in the process of extraction, such as saw tooth, projection and local deformation. Previous experiments have shown that the calculated results are better on k = 5.

## 2.2 Method for Gesture Preprocessing

In order to recognize the gestures, the images of the gesture need to be preprocessed. The pretreatment method is as follows. Firstly, the gesture image is captured, as shown in Figure 1(a). Due to the nature of the YCbCr color space of brightness, the chroma can be separated to overcome certain illumination interference. This is used for skin color segmentation [13]. Secondly, the gesture region that contains the skin region is extracted (as shown in Figure 1(b)). Because of the influence of the background, the segmentation results will contain different levels of noise. Thirdly, in order to reduce the noise, two-value segmentation of the image is done by removing small connected domains and hole filling, and the palm area is obtained (Figure 1(c)). Finally, the fingertip location method based on contour curvature is used to obtain the palm region, as shown in Figure 1(d).

(a) original image (b) skin color segmentation (c)gesture region (d)palm area

**Figure 1** Gesture preprocessing.

## 2.3 Construct the Normalized Palm Contour Sequence

Based on the idea of boundary tracking, the palm contour sequence is constructed. Firstly, the Sobel operator is used to obtain the external edge information of the palm, which is shown in Figure 2(a). The outer edge of the palm is extracted; its construction steps are as follows:

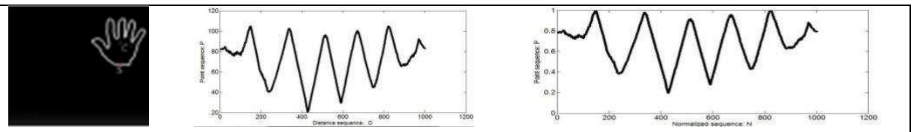
**Step 1**: Calculate the centroid point of the gesture, marked as point C.

**Step 2**: Calculate the coordinates of the farthest distance from point C to the outer edge of the wrist. This is denoted as point S (Figure 2(a)).

**Step 3**: According to the clockwise direction, the distance between each point in the gesture contour and point C is calculated. Among them, point S is the starting point. The calculated distance values are denoted as sequence D. According to the order of records, these points are recorded as point sequence P.

**Step 4**: Sequence P is the abscissa and the sequence D is the ordinate. This is expanded as shown in Figure 2(b).

**Step 5**: Sequence D is normalized. That is, all the distance values are mapped to 0 or 1. The normalized palm contour sequence is denoted as sequence N, which is shown in Figure 2(c).



(a)palm contour (b) palm contour sequence (c) normalized palm contour sequence

**Figure 2** Construction of palm contour sequence.

# 3 Recursive Palm Contour Sequence Model

## 3.1 Definition of Gesture Contour Sequence

Given a sequence of samples $E_i$ and a discrete class label $T_i$ [14], the gesture contour sequence is defined as follows:

$$E_i = e(i_1), e(i_2), ..., e(i_n) \tag{2}$$

Where the variable $E_i$ is the sequence characteristic of n-dimensional edge vector i corresponding to the gesture. The variable $e(i_t)$ is the characteristic point of $E_i$ at the corresponding position $i_t$. The label $T_i$ ($T_i \in \{1,2,...,N\}$) is a gesture type label. After preprocessing the gestures G6, G2 and G7, the binary images and the corresponding gesture sequence are got, as shown in Figures 3 and 4.

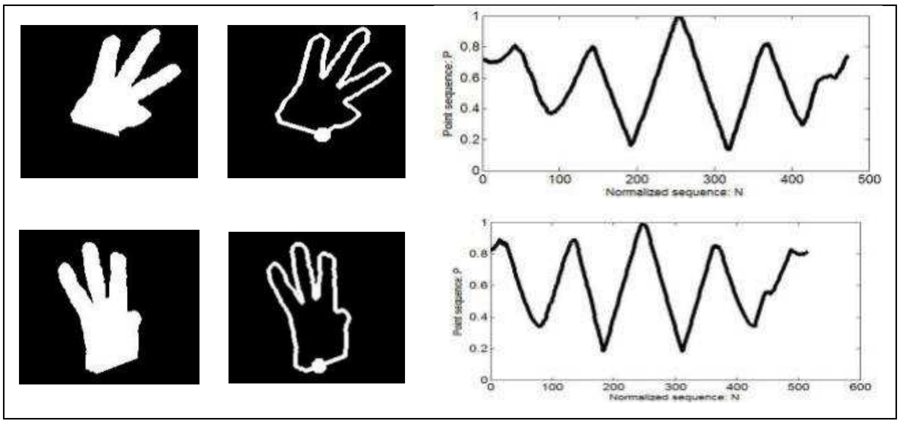



**Figure 3** G6 palm contour sequence.

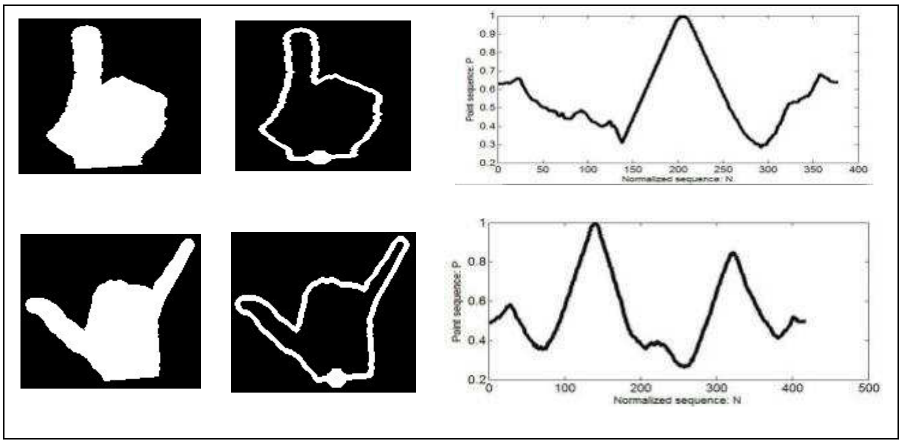



**Figure 4** Palm contour sequence of G2 and G7.

## 3.2 Construct the Recursive Model of Palm Contour Sequence

For the analysis of the similarity degree between two gesture contour series, a similar distance metric is adopted [15]. The common measurement methods include Hamming distance, Euclidean distance, dynamic time warping (DTW) and so on. Among them, DTW is the most commonly used method. Because of the orientation of the gesture and the location of the wrist, the recognition algorithm can easily make mistakes. In this paper, the recurrence plot is introduced to deal with this problem.

A recurrence plot can reflect repeating patterns in a time series. It is often used to analyze the time series of chaotic, non-stationary and periodic characteristics. The recurrence plot has good internal characteristics in short time series data and has good application in the fields of climate change, stock market analysis, DNA sequence structure analysis, and so on [16].

Recursive graph compression can guarantee the consistency of an image in rotation, translation and scaling. Because of the short edge sequence of the gesture, the number of fingertips and fingers can be seen as a repeat pattern [17]. Therefore, a recursive model is proposed to express the repeated sequence of a gesture contour. This idea is used to solve the problem of robust matching of the starting point of the gesture contour sequence and achieving gesture similarity detection.

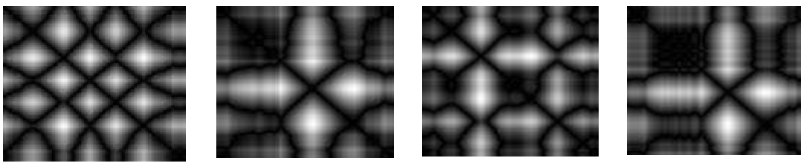
In order to observe the trace of the gesture sequence, according to the given integer $n > 0, \tau > 0$, contour sequence $\mathrm{E}_i$ is transformed into a recursive graph using Eq. (3).

$$R = r_{i,j} = \theta\left(\varepsilon - \left\|e(i_k) - e(i_m)\right\|\right) \qquad i_k, i_m = 1,2,\cdots n \tag{3}$$

Where the variable n is the number of states experienced by the spatial sequence. The variables $e(i_k)$ and $e(i_m)$ are the values of the spatial sequence observed at the $i_k$ and $i_m$ sequence positions. The symbol $\|\cdot\|$ represents the distance between two observation points. The symbol $\varepsilon$ represents a threshold. The symbol $\theta$ is a Heaviside step function, which is defined as in Eq. (4):

$$\theta(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases} \tag{4}$$

Figure 5 shows the contour sequence recursive models of four gestures (G5, G2, G7, G2). The internal structures of the four gestures can be observed directly and the four recursive models are symmetrically distributed on both sides of the diagonal. Figures 5(d) and 5(b) represent the same gesture in different images generated by the contour sequence of the recursive map. It can be seen from the figure that they have a similar internal spatial structure.



(a) G5 (b) G2 (c) G7 (d) G2

**Figure 5** Recursive graphs of different hand gestures.

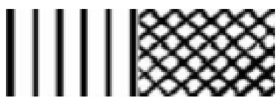
In a word, the recursive graph of the gesture contour sequence provides a good division of the gesture. Because the recursive model transforms the unequal gesture contour sequence into two-dimensional graphs of the same size, this method solves the problem of unequal length and initial point matching in the similarity detection of gesture edge sequences.

## 4 Gesture Similarity Detection

Because each gesture can obtain a different recursive image, it can be judged whether two images represent the same gesture by comparing the similarity between them. It can be seen from Figure 5 that the image is symmetrical in both ends of the diagonal. Therefore, the comparison of the triangular part can be used to determine the type of gesture. In this paper, a new triangular image texture similarity detection algorithm is proposed. Compared with the whole image, the efficiency of the proposed algorithm will be doubled in terms of the reduction of computation.

### 4.1 Image Texture Feature Selection

Hung [18] has proposed a texture classification method based on K-Views. The idea behind this method is as follows. Figure 6(a) shows two images with different textures. Figure 6(b) contains the gray values of the pixels in the corresponding positions. A part of two images with different textures is selected as the research object. The characteristics of both images are used to find whether the images belong to the same class.



(a) Binary images

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0
```

(b) Pixel gray values for both images in (a)

**Figure 6** Images with different textures.

There are many statistical features of images. Among them, histogram, entropy, mean value and standard deviation are widely used. Because these four features are simple and convenient to implement and have the characteristics of rotation in variance, they can be used to distinguish the images of different gestures. In order to solve the instability of image recognition in the process of image rotation, these feature values are introduced in the proposed algorithm.

1. *Histogram*

   The histogram is used to count the number or probability of each gray level in an image. It can be represented as a two-dimensional graph, where the abscissa represents the gray level or range of each pixel in the image, and the ordinate is the number or probability of each pixel in each gray level or gray level range.

2. *Entropy*

Entropy is a measure of the amount of information in an image. It represents the degree of heterogeneity or complexity of the texture in the image. Its definition is shown in Eq. (5).

$$H = -\sum_{i=1}^{N} p_i * \ln p_i \tag{5}$$

The symbol $p_i$ represents the proportion of pixels $i$ in the gray values of the image.

3. *Mean*

Mean represents the average gray value. Its definition is shown in Eq. (6):

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6}$$

4. *Standard deviation*

Standard deviation shows the fluctuation of the measured value to the mean. Its definition is shown in Eq. (7):

$$S = \left[\frac{1}{N-1}\left(\sum_{i=1}^{n} x_i^{\,2} - n\bar{x}^2\right)\right]^{\frac{1}{2}} \tag{7}$$

## 4.2 Normalized Image Feature Vector

The four features are combined to form a normalized feature vector, which has the rotation invariant feature. The definition of the feature vector is shown in Eq. (8):

$$x = \{x_1, x_2, \ldots, x_c\} \tag{8}$$

Among them, the symbol c represents the feature space. In the algorithm, the value of c is set to 4. In order to keep the generality, the feature vector x is normalized to get the normalized feature vector as is shown in Eq. (9):

$$x_{nc} = \begin{cases} 0 & , \text{if } x_{nc} < (x_{nc})_{\min} \\ 1 & , \text{if } x_{nc} > (x_{nc})_{\max} \\ \dfrac{x_{nc} - (x_{nc})_{\min}}{(x_{nc})_{\max} - (x_{nc})_{\min}} & , \text{else} \end{cases} \tag{9}$$

Where, the definitions of $(x_{nc})_{\min}$ and $(x_{nc})_{\max}$ are Eqs. (10) and (11) respectively:

$$(x_{nc})_{\min} = \left\{ \min_{n=1}^{S} x_{n1}, \min_{n=1}^{S} x_{n2}, \ldots, \min_{n=1}^{S} x_{nC} \right\} \tag{10}$$

$$(x_{nc})_{\max} = \left\{ \max_{n=1}^{S} x_{n1}, \max_{n=1}^{S} x_{n2}, \ldots, \max_{n=1}^{S} x_{nC} \right\} \tag{11}$$

Where the variable S represents the sum of the squares of all sample images.

### 4.3 Definition of Classification Probability

The definition of the probability that the current image belongs to gesture $j$ is shown in Eq. (12). The variable $S_j$ represents the probability.

$$S_j = \sum_{i=1}^{V} (s_{i,j})^2 \tag{12}$$

The variable $V$ represents the number of squares in the current image. The variable $s_{i,j}$ represents the probability that the first $i$ block in the current image belongs to gesture $j$. It is equivalent to the similarity between the feature vectors of the first $i$ squares and the characteristic vectors of gesture $j$ [19]. There are many methods to measure vector similarity. In the UTTRG-SGR algorithm, the Euclidean distance is used as the measure of scale. Its specific definition is shown in Eq. (13):

$$s_{ij} = \frac{1}{|d_{ij}|} \tag{13}$$

Where the variable $d_{ij}$ represents the minimum Euclidean distance value of each eigenvalue between the first $i$ square and gesture $j$.

### 4.4 Gesture Recognition Process

The gesture recognition algorithm implementation steps are as follows:

**Step 1:** From the (0,0) point of the identified image, select the appropriate size of the box. In the upper triangular part of the image, the box set is extracted.

**Step 2:** The eigenvalues of these squares are calculated as the eigenvalues of the corresponding gesture classes.

**Step 3:** The obtained feature values are mapped into the normalized eigenvalues.

**Step 4:** Compared with the existing gesture categories, the probability of the image belonging to the gesture $j$ is calculated. It is represented by $S_j$ .

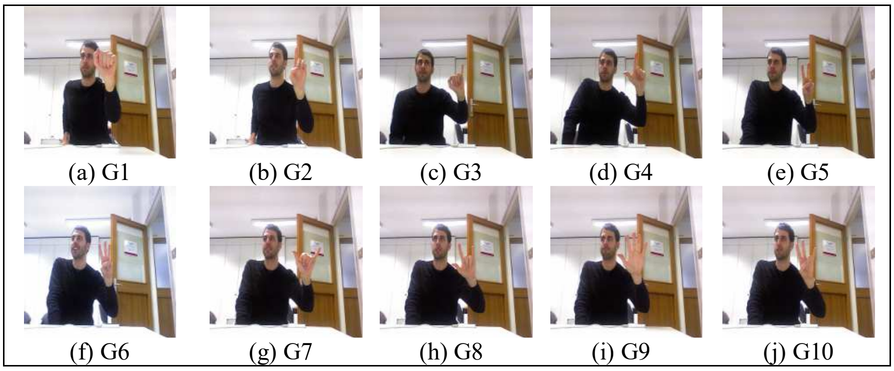
**Step 5:** When the probability value is the largest, the image is classified as the gesture.

## 5 Experimental Results and Analysis

The accuracy of gesture classification was simulated by using the public gesture data set provided by University of Padua. Data sets were collected from 14 individuals with 10 different gestures, and each gesture was repeated for 10 times to get a total of 1400 hand gestures.

The experimental environment was as follows: (1) Memory: 8G RAM. (2) CPU: i5-5200 Intel. (3) Software: Matlab2014.

The 10 different static gestures of a person are defined as shown in Figure 7.



(a) G1 (b) G2 (c) G3 (d) G4 (e) G5

(f) G6 (g) G7 (h) G8 (i) G9 (j) G10

**Figure 7** 10 types of static hand gestures.

The process of the gesture recognition algorithm is divided into two stages: training and recognition [20]. The training phase is divided into 3 steps:

**Step 1:** The gesture image is processed and the binary image is obtained.

**Step 2:** Get the gesture contour sequence.

**Step 3:** Construct a recursive model of the gesture contour sequence.

In the recognition phase, the normalized distance between the images is calculated by means of the triangular image texture similarity detection algorithm. The highest probability gesture is used as the recognition result. Using the method of cross examination, 8 people were randomly selected at each time. Each gesture contained a total of 80 samples. 600 images were used for the experiment and the other 800 images were used for training.

The algorithms in reference 10 and 11 are very representative. In order to verify the performance of the proposed algorithm, the experimental results are compared with the two algorithms. The results of gesture recognition are shown in Figure 8.

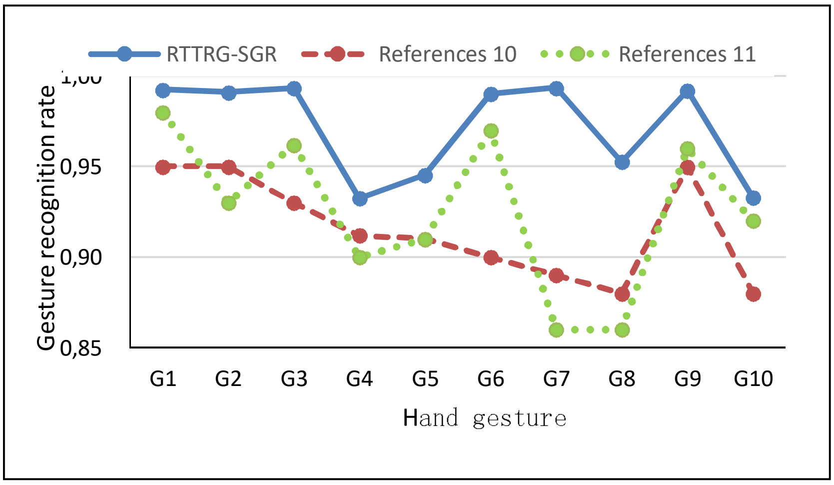



**Figure 8** Gesture recognition rate results.

According to the analysis of each gesture, the recognition accuracy of the proposed algorithm for gesture 1, gesture 2, gesture 3, gesture 6, gesture 7 and gesture 9 was higher, and the accuracy of the UTTRG-SGR algorithm was close to 100%. Other gestures, due to the same number of extended fingers, may lead to confusion in gesture classification. This may lead to an error in the starting point coordinates and the decline of the accuracy of gesture classification. It can be seen that the accuracy of the UTTRG-SGR algorithm is high and it has a high universality for the rotation, translation and scaling. The average accuracy of the proposed algorithm is 97.2%. In a word, the recognition rate of the UTTRG-SGR algorithm is higher than that of the other two algorithms.

The execution time of the three algorithms is shown in Figure 9. As can be seen from the figure, the execution time of the proposed algorithm is between 1 and 2 seconds, which is shorter than the other two algorithms. Therefore, the UTTRG-SGR algorithm is more efficient.

Another data set from University of Padua includes 13035 gesture images. Many of them are the same gestures in rotation or translation. In order to verify the stability of the algorithm, 5200 images were selected randomly. These images were from 5 different gestures, as shown in Figure 10. The average accuracy of the experimental results was 97.5%.

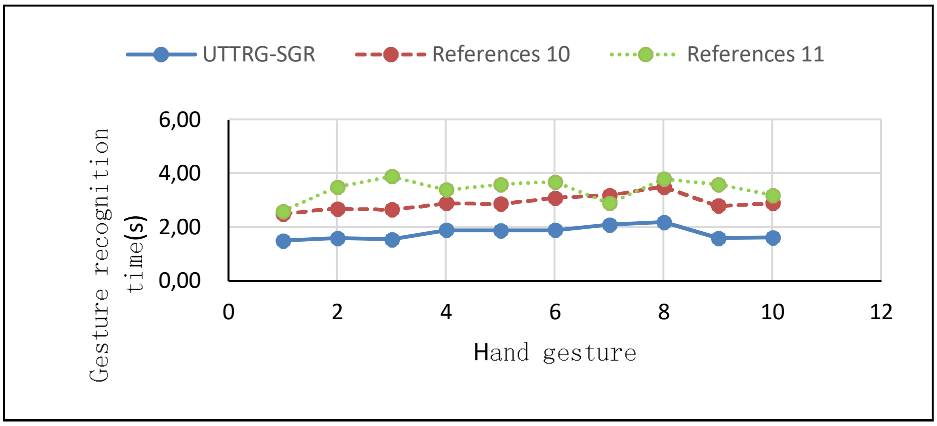



**Figure 9** Execution time comparison.

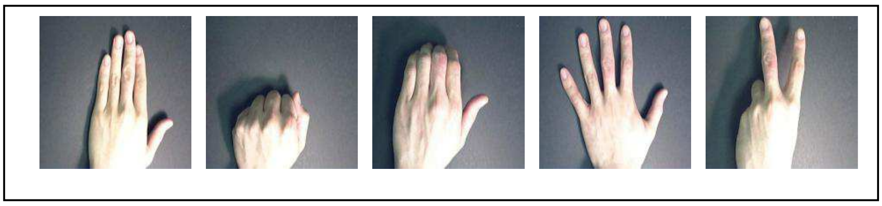


**Figure 10** 5 type gestures.

In summary, the UTTRG-SGR algorithm has better accuracy and efficiency. When the gestures are in rotation, translation, or scaling, the algorithm still has very good performance.

## 6 Conclusion

In this paper, a new image texture algorithm for gesture sequence was presented. In this algorithm, firstly, the contour curvature is used to determine the gesture contour. Then the gesture contour is converted to a sequence model with time varying rules. Secondly, a similarity detection method of texture image was proposed from the view of the spatial domain. In order to make full use of the contour information of the gesture image, a new gesture feature extraction method was put forward based on the spatial domain, which deals with the problems of the complicated feature extraction method and finite gesture characteristics. Finally, a new image texture recognition algorithm was proposed to detect the similarity between images. The experimental results showed that the proposed algorithm achieved high robustness with 97.2% recognition accuracy on a data set of 10 kinds of static gestures provided by the University of Padua. Its operation time was also shorter than that of existing methods. The other results showed that the proposed algorithm is robust to rotation, translation and scaling of gestures. All in all, it has superior performance and is worthy of promotion.

The proposed algorithm requires that the palm area is clear and identifiable. As a result, the algorithm has some shortcomings. These are mainly manifested as follows: (1) when the gesture and the background image are too close, the algorithm cannot correctly identify the gesture. (2) When gestures are too similar, the result is not very good. These issues will be the focus of future research.

## Acknowledgements

This work was supported by the Research on Henan Province Natural Science Fund Project (142300410413, 152300410219) and the Education Department of Henan province science and technology research project (12A520033).

## References

[1] Hogreve, S., Kaczmarek, S., Adam, I., Franz, L., Döllen, T., Paulus, H., Reinkemeyer, V., Tracht, K., *Controlling and Assisting Manual Assembly Processes by Automated Progress and Gesture Recognition*, Applied Mechanics and Materials, **840**, pp. 50-57, 2016.

[2] Akoum, A. & Al Mawla, N., *Hand Gesture Recognition Approach for ASL Language using Hand Extraction Algorithm*, Journal of Software Engineering and Applications, **8** (8), pp. 419-430, 2015.

[3] Badi, H., Hussein, S.H. & Kareem, S.A., *Feature Extraction and Ml Techniques for Static Gesture Recognition*, Neural Computing and Applications, **25**(3-4), pp. 733-741, 2014.

[4] Raheja, J.L., Minhas, M., Prashanth, P., Shah, T. & Chaudhary, A., *Robust Gesture Recognition Using Kinect: A Comparison between DTW and HMM*, Optik-International Journal for Light and Electron Optics, **126**(11-12), pp. 1098-1104, 2015.

[5] El Kashef, N. Hasan, Y.F., Mahar, K. & Fahmy, M.H., *Philosophy of Gustatory Perception Strategy for Hand Gesture Recognition*, Kybernetes, **44** (3), pp. 451-459, 2015.

[6] Ribó, Alba., Warchol, D. & Oszust, W., *An Approach to Gesture Recognition with Skeletal Data using Dynamic Time Warping and Nearest Neighbour Classifier*, International Journal of Intelligent Systems and Applications, **8** (6), pp. 1-8, 2016.

[7] Singha, Joyeeta & Laskar, Rabul Hussain, *ANN-Based Hand Gesture Recognition Using Self co-articulated Set of Features*, IETE Journal of Research, **61**(6), pp. 597-608, 2015.

[8] Izuta, R., Murao, K., Terada, T. & Tsukamoto, M., *Early Gesture Recognition Method with an Accelerometer*, International Journal of Pervasive Computing and Communications, **11** (3), pp. 270-287, 2015.

[9] Artyukhin, S.G. & Mestetskiy, L.M., *Dactyl Alphabet Gesture Recognition in a Video Sequence using Microsoft Kinect*, ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **XL-5/W6** (1), pp. 83-86, 2015.

[10] Dardas, N.H. & Georganas, N.D., *Real-time Hand Gesture Detection and Recognition using Bag-of-Features and Support Vector Machine Techniques*, IEEE Trans. on Instrumention and Measurement, **60**(11), pp.3592-3607, 2011.

[11] Priyal, P.S. & Bora, P.K., *A Robust Static Hand Gesture Recognition System Using Geometry Based Normalizations and Krawtchouk Moments,* Pattern Recognition, **46**(8), pp. 2202-2219, 2013.

[12] Yang, X.W., Feng Z.Q., Huang, Z.Z. & He, N.N., *A Gesture Recognition Algorithm using Hausdorff-Like Distance Template Matching Based on the Main Direction of Gesture*, Applied Mechanics and Materials, 713-715, pp. 2156-2159, 2015.

[13] Kumawat, V. & Agarwal, S., *Robot Navigation Control with Contact Less Hand Gesture Recognition*, INROADS- An International Journal of Jaipur National University, **5** (1s), pp. 104-108, 2016.

[14] Kim, K. & Choi, H.-I., *Online Hand Gesture Recognition using Enhanced $N Recogniser Based on a Depth Camera*, Int. J. of Computational Vision and Robotics, **6** (3), pp. 214-222, 2016.

[15] El-Shazly, E.H., Abdelwahab, M.M., Shimada, A. & Taniguchi, R., *Early Gesture Recognition with Adaptive Window Selection Employing Canonical Correlation Analysis for Gaming*, Electronics Letters, **52**(16), pp. 1379-1381, 2016.

[16] C. Yang, Ku, B., Han, D.K. & Ko, H., *Alpha-Numeric Hand Gesture Recognition Based on Fusion of Spatial Feature Modelling and Temporal Feature Modelling*, Electronics Letters, **52**(20), pp. 1679-1681, 2016.

[17] Hassan, A., Shafi, M. & Khattak, M.I., *Multi-Touch Collaborative Gesture Recognition Based User Interfaces as Behavioral Interventions for Children with Autistic Spectrum Disorder: A Review*, Mehran University Research Journal of Engineering and Technology, **35** (4), pp. 543-560, 2016.

[18] Hung, C.C., Karabudak, D., Pham, M. & Coleman, M., *Experiments on Image Texture Classification with K-Views Classifier*, *Markov Random Fields and Co-occurrence Probabilities, 2004 IEEE International Geoscience and Remote Sensing Symposium,* Anchorage, AK, **6**, pp. 3818-3821, 20-24 September, 2004.

[19] Kim, D., Lee, J., Yoo, H., Kim, J. & Sohn, J., *Vision-Based Arm Gesture Recognition for a Long-Range Human-Robot Interaction*, Journal of Supercomputing, **65**(1), pp. 336-352, 2013.

[20] Patel, Deval G., *Point Pattern Matching Algorithm for Recognition of 36 ASL Gestures*, International Journal of Science and Modern Engineering, **7**(1), pp. 24-28, 2013.