Gesture Recognition Analysis Based On Monocular Vision Relying On Human-Computer Interaction Filtering Algorithm

Qin Liu*
School of Business Administration, Shandong Institute of Commerce & Technology, Jinan 250103, Shandong, China

*Corresponding author e-mail: liuqin@sict.edu.cn

Abstract. The progress of vision-based gesture recognition technology has introduced a brand new way of human-computer interaction (HCI). However, due to the complexity of feature extraction, gesture recognition based on monocular vision has become a research hotspot and bottleneck. In this paper, static gestures are combined with the CSS feature description to extract the skin-color model based on the HCI filtering algorithm first. Subsequently, the fusion feature is established. Finally, feature recognition is performed based on the HCI filtering algorithm to implement gesture recognition analysis based on monocular vision. The practice has demonstrated that the proposed algorithm has a relatively high recognition rate for complex gestures with the sign language of high local similarity.

Keywords: Human-Computer Interaction (HCI) Filtering Algorithm, Gesture Recognition, Monocular Vision

Gesture recognition is an essential interaction method in HCI. It is mainly used to detect, track, and recognize user gestures through video input devices (cameras, etc.) to understand human intentions via computer [1]. From the initial application of sign language recognition to the innovation of somatosensory games, gesture recognition technology has become increasingly sophisticated. Currently, gesture recognition is also closely integrated with other fields, including medical support, smart cars, and robots [2].

Vision-based computer recognition is mainly divided into recognition based on the image itself and recognition based on the depth image. The main principle is to separate the image through the threshold and extract the required sample features as needed. Human hands are complex deformable bodies. Gestures have ambiguity, polymorphism, and time and space differences. Hence, the selection and extraction of features is a bottleneck in the field of gesture recognition. Some scholars have introduced optical marks into the recognition system to improve the accuracy of recognition, and they have also achieved good results. Although external devices such as data gloves and visual signs have improved recognition stability and accuracy, they have become a challenging research topic [3-5]. In general, gestures can be divided into static gestures and dynamic gestures. The feature extraction of a model is to build a mathematical model of the human hand based on empirical knowledge.
Subsequently, the parameters of the model are estimated based on the features of the gesture. Finally, the template matching method is used for recognition and classification. This method can handle more complex gestures [6]. The disadvantage is that it requires a large amount of calculation. When the image resolution is low, it is not easy to estimate the parameters of the model. The apparent feature of the image is to use the geometric features of the palm and fingers of the gesture contour on the image to describe the gesture feature.

In this paper, this paper attempts to integrate static gestures are combined with the CSS feature description based on the HCI filtering algorithm to establish special feature gestures for easy extraction.

1. Gesture Segmentation of HCI Filtering Algorithm

In gesture segmentation, the contour of gestures is detected from the complex background. Currently, more sophisticated segmentation algorithms include the frame difference algorithm and the skin-color model segmentation method. The skin-color segmentation method has a relatively small amount of calculation and a simple model, so this paper uses the skin-color model segmentation method.

In general, gestures can be divided into static gestures and dynamic gestures. This article mainly reviews the research of vision-based dynamic gesture recognition in recent years, focusing on the four aspects of gesture segmentation, gesture tracking, feature extraction, and recognition algorithms. The basic process of the gesture-based recognition system is shown in Figure 1.

![Figure 1. Block diagram of the general gesture recognition process](image)

For invariant moments, it is lack of local description features, while CSS feature descriptors have no overall features. In this paper, two features are merged, the gesture is described from both the overall and the local aspects, and then the new features are classified by the artificial neural network to achieve an excellent classification effect. Figure 2 shows the gesture recognition flow of the HCI filtering algorithm.

![Figure 2. Flow chart](image)
1.1. Skin-color Model
The skin-color segmentation model uses the clustering of skin-color in the color space to separate the region of interest from the complex background environment. Since skin-color has different clustering characteristics in different color spaces, to obtain a good segmentation effect, a suitable color space must be selected. Currently, there are three commonly used color spaces: RGB color space, HSV color space, and YCrCb color space. Through a large number of experiments, it is proved that in YCrCb space, skin-color is less affected by brightness, and skin-color clustering characteristics are superior. Hence, the YCrCbir space is selected in this paper as the color space for gesture segmentation.

1.2. Extraction of Gesture Contour
In the experiment, we use a monocular camera to obtain an image containing gesture information. The image resolution is 320×240. The complete gesture contour can be obtained through the following steps:

Step 1: Use equation (1) to convert the image from RGB space to YCrCb space:

\[
\begin{align*}
Y &= 0.257 \times R + 0.564 \times G + 0.098 \times B + 16 \\
Cb &= -0.148 \times R - 0.291 \times G + 0.439 \times B + 128 \\
Cr &= 0.439 \times R - 0.368 \times G - 0.071 \times B + 128
\end{align*}
\] 

(1)

Step 2: Conduct threshold segmentation on the YCrCb image acquired to obtain a binary image of the gesture. In the experimental environment, the value range of Cr and Cb: 133≤Cr≤183, 78≤a≤131.

Step 3: To remove noise and interference, perform filtering and graphics processing on the binary image after threshold segmentation.

Step 4: Extract the contour of the binary image obtained in Step 3.

Step 5: Remove the interference of the non-skin area, we set the contour point threshold T. When the contour points are less than the threshold r, the area is considered to be not a gesture contour and filled with black, and finally an image with only the gesture contour is obtained.

In feature extraction, the region of interest in the input sample is converted into a collection of feature vectors. In gesture recognition, the extracted feature should contain the relevant information of the gesture to be measured. It should be expressed in a compact form as the identification of the gesture and distinguished from other gestures. Combined with most of the existing literature, dynamic gesture features mainly include global features, local features, and fusion features.

1.3. Global Features
Global features are often used to reflect the overall attributes of image sequences, which can be described by motion history maps, motion energy maps, and space-time shapes. Compared with other distance measurement methods, only the outer contour of the finger is considered in the proposed algorithm, while the palm area is not included. It represents the contour of the hand through a set of time series curves that record the relative distance between the vertex of each contour and the center of the palm. Each finger corresponds to a certain segment of the curve. For example, the difference between different gestures is determined based on the Super Pixel Land Movement Distance (SPEMD) algorithm, where hand shape and texture information are described in the form of superpixels. This method has robust performance for the rotation, translation, and zooming of gestures.

1.4. Local Features
This type of feature extracts the local feature points to characterize dynamic gestures in the image sequence effectively and then performs statistical modeling of various attributes of these feature points to achieve the description of dynamic gestures. For example, the HON4G (Histogram of Oriented 4D Normals) descriptor is used to describe the depth sequence, which uses the histogram of the normal
surface direction in the 4D space of time, depth and space coordinates to capture motion and geometric data. The experimental results on the MSR Gesture 3D dataset have verified the superiority of the descriptor.

1.5. Fusion Features
For some specific gestures, a single feature can usually give excellent recognition results. However, its application range is small, and its stability is average. To improve stability and applicability, scholars have considered a scheme that combines multiple features. First, the normalized motion energy vector is divided into a group of segments, and the depth video is divided into a group of grids using the corresponding frame index; subsequently, the HOG feature is used to extract the local texture information and motion information of the gesture from the depth sequence.

2. Extraction of Gesture Contour Features

2.1. CSS Feature Descriptor
The CSS feature describes the shape feature of a gesture by the curvature of each point on the contour of gestures. The curvature distribution of each point on the contour of different gestures is different. The CSS feature descriptor is to find the zero-crossing points of the gesture contour in the scale-space of the image and use the extreme point positions in these zero-crossing combinations and the set of corresponding spatial scale information as the descriptor.

(1) Curvature calculation

Use arc length $\mu$ to parameterize the curve:

$$ L(\mu) = (x(\mu), y(\mu)) $$

(2)

Then the curvature of each point on the curve can be calculated by the following equation:

$$ g(\mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \frac{e^{-\frac{\mu^2}{2\sigma^2}}}{\sigma} $$

(3)

Where $g(\mu, \sigma)$ represents a Gaussian convolution kernel with width $\sigma$, $x(\mu, \sigma)$ and $y(\mu, \sigma)$ represent obtained by Gaussian convolution of curve $L(\mu)$, $\otimes$ represents convolution, $\dot{x}(\mu, \sigma)$ and $\ddot{x}(\mu, \sigma)$ represent the first derivative and the second derivative of $x(\mu, \sigma)$, respectively.

(2) CSS descriptor generation algorithm

Based on the above curvature calculation equation, the curvature of the obtained gesture contour is calculated, and the CSS feature descriptor is obtained based on the following steps:

Step 1: Parameterize the obtained gesture contour to obtain $L(\mu)$

Step 2: Identify the curvature of each point of the curve based on equation (3), and obtain the contour curvature of the gesture at scale 6

Step 3: Determine whether there are extreme curvature points in the curvature sequence at scale $\sigma$. If so, go to step 4, especially go to step 5.

Step 4: Record the position and scale of the extreme point of curvature, denoted by $(\mu, \sigma)$, increase the scale $\sigma = \sigma + 1$, go to step 3.

Step 5: Plot the points obtained in step 4 on the $(\mu, \sigma)$ plane to obtain the scale-space image CCSI.

Under the condition of the uniform discretization parameter $\mu$, the coordinate set of all local extrema in CCSI is the CSS shape descriptor, that is:

$$ F_{CSS} = \{(\mu_i, \sigma_i) | i = 1, 2, \cdots, N\} $$

(4)

2.2. Moment Invariant Features
Moment invariance was first proposed by Hu et al. in 1962. The moment invariant addresses were combined linearly to obtain moments with scale invariance, translation invariance, and rotation invariance.

For a two-dimensional digital image \( f(x,y) \), the corresponding \( p+q \) moments are:

\[
m_{pq} = \sum_x \sum_y f(x,y)x^py^q, \quad p, q = 0, 1, 2, \ldots
\]

The center distance \( u_{pq} \) is standardized to obtain the following center moments:

\[
\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}} = \frac{p+q}{2} + 1, \quad p+q = 2, 3, \ldots
\]

The 7 Hu invariants acquired above are used to describe the contour feature of the gesture, and express it as the following form descriptor:

\[
F_{Hu} = \{M_1, M_2, M_3, M_4, M_5, M_6, M_7\}
\]

2.3. Feature Fusion

The spatial fairy rate is calculated to obtain CSS shape descriptor \( FCSS \), and the Hu invariant moment descriptor \( F_{Hu} \) with 7 features is obtained by Hu invariant moments. However, different distance measurement methods are used for CSS shape descriptors and Hu invariant moment descriptors. The dimensions are not uniform, and the dimensionality of CSS shape descriptors is uncertain. Direct splicing can result in imbalance. Hence, their direct combination is not feasible. Descriptor fusion can be used to reduce the effect of direct fusion by normalizing and weighting features, as follows:

Step 1: Normalize the CSS descriptor and Hu invariant moment descriptor to make the size between 0 and 1.

Step 2: Conduct linear weighting on the normalized distance to obtain new features:

\[
F = a \times FCSS + b \times F_{Hu}
\]

Where \( a \) and \( b \) are the weights, which should be determined during the experiment. \( FCSS \) is the CSS descriptor, and \( F_{Hu} \) is the Hu feature descriptor.

To determine the values of linear weighting coefficients \( a \) and \( b \) in equation, different values of \( a \) and \( b \) were tested. The results are shown in Table 1. When the value of \( a \) is small, the proportion of CSS features is relatively large. The overall feature is missing, and the recognition rate is relatively low. With the increase in the value of \( a \), the overall recognition rate increases. When \( a \) increases to 0.4, the recognition rate reaches the maximum value. As \( a \) continues to increase, the proportion of Hu invariant moment features gradually increases. Due to the lack of local description, the recognition rate begins to decrease. Based on the above analysis, this experiment takes \( a=0.4 \), \( b=0.6 \), and the results are shown in the fifth row of Table 1.

| \( a \) | \( b \) | Recognition rate |
|---|---|---|
| 0.1 | 0.9 | 84.6% |
| 0.2 | 0.8 | 85.1% |
| 0.3 | 0.7 | 88.7% |
| 0.35 | 0.65 | 89.3% |
| 0.4 | 0.6 | 92.4% |
| 0.45 | 0.55 | 90.6% |
| 0.5 | 0.5 | 89.6% |
| 0.55 | 0.45 | 88.7% |
3. Analysis and Comparison of Experimental Results

For the 30 sign language letters in the alphabet, gesture images were collected from 6 different experimental subjects using a camera. Each gesture was collected 20 times. A total of 3,600 samples were obtained, of which 2,700 were used for sample training, and 900 were used for training. The experimental results of 900 test samples are shown in Table 2.

Table 2. Different gesture recognition rate

| Gesture | Recognition rate | Gesture | Recognition rate | Gesture | Recognition rate |
|---------|------------------|---------|------------------|---------|------------------|
| A       | 100%             | K       | 90.2%            | U       | 93.2%            |
| B       | 96.4%            | L       | 86.3%            | V       | 93.1%            |
| C       | 93.2%            | M       | 80.4%            | W       | 90.5%            |
| D       | 100%             | N       | 76.5%            | X       | 80.2%            |
| E       | 93.1%            | O       | 86.5%            | Y       | 90.1%            |
| F       | 93.1%            | P       | 83.2%            | Z       | 93.2%            |
| G       | 96.4%            | Q       | 90.1%            | ZH      | 86.5%            |
| H       | 90.4%            | R       | 93.2%            | CH      | 90.6%            |
| I       | 93.2%            | S       | 90.1%            | SH      | 83.2%            |
| J       | 86.1%            | T       | 86.4%            | NG      | 86.6%            |

Nine hundred test samples were tested using the method in this paper, and good experimental results were obtained. The recognition rate of some gestures reached 100%, and the overall recognition rate was 92.4%.

To verify the effectiveness of the proposed method, 900 samples (30 samples for each gesture) were tested using Hu invariant moment features and CSS features. The results suggest that for simple gestures, using Hu invariants or CSS features alone can achieve a higher recognition rate. For example, for gesture A, the number of correct recognition using two features alone reaches 28 and 29, respectively. However, for some complex gestures or gestures with small local discrimination, the recognition rate of this method is much higher than that of using one feature alone. Promote. For example, for gesture M and gesture N, the local similarity of these two gestures is very high. The correct recognition numbers of CSS features are 20 and 19, respectively. The Hu invariant moment features are used alone, and the recognition rates are 19 and 18, respectively. The number of correct recognition using CSS feature and Hu invariant moment feature reached 24 and 23, respectively.

4. Conclusions

The progress of vision-based gesture recognition technology has led to a brand new way of HCI. Users can use computers to complete interactive functions with bare hands, which has successfully freed users from the shackles of external devices such as data gloves and optical signs with improved flexibility and naturalness of HCI. In this paper, static gestures are combined with the CSS feature description based on the HCI filtering algorithm to perform gesture recognition analysis based on monocular vision. The practice has demonstrated that the proposed algorithm has a relatively high recognition rate for complex gestures with sign language of relatively high local similarity.

References
[1] Qiu K, Qin T, Gao W, et al. Tracking 3-D Motion of Dynamic Objects Using Monocular Visual-Inertial Sensing. IEEE Transactions on Robotics, 2019, 35(4):799-816.
[2] Dong X, Dong X, Dong J, et al. Monocular Visual-IMU Odometry: A Comparative Evaluation of Detector–Descriptor-Based Methods. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(6):2471-2484.
[3] Sheng B, Li P, Zhang Y, et al. GreenSea: Visual Soccer Analysis Using Broad Learning System. IEEE Transactions on Cybernetics, 2020, 3(9):1-15.

[4] Lu Q, Xu W, Zhang H, et al. ElectricVIS: visual analysis system for power supply data of smart city. Journal of supercomputing, 2020, 76(2):793-813.

[5] B L L A, D S H C, E Y J B D, et al. Natural-scene-based Steady-state Visual Evoked Potentials Reveal Effects of Short-term Monocular Deprivation - ScienceDirect. Neuroscience, 2020, 435(4):10-21.

[6] A X D, B M A, C S B, et al. Patch-based adaptive weighting with segmentation and scale (PAWSS) for visual tracking in surgical video. Medical Image Analysis, 2019, 57(5):120-135.