**Journal of Software Engineering and Applications, 2016, 9, 103-111**
Published Online April 2016 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2016.94009

# Hand Gesture Recognition Using Appearance Features Based on 3D Point Cloud

**Yanwen Chong, Jianfeng Huang, Shaoming Pan**

State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China
Email: ywchong@whu.edu.cn

Received 6 March 2016; accepted 15 April 2016; published 18 April 2016



## Abstract

**This paper presents a method for hand gesture recognition based on 3D point cloud. Digital image processing technology is used in this research. Based on the 3D point from depth camera, the system firstly extracts some raw data of the hand. After the data segmentation and preprocessing, three kinds of appearance features are extracted, including the number of stretched fingers, the angles between fingers and the gesture region's area distribution feature. Based on these features, the system implements the identification of the gestures by using decision tree method. The results of experiment demonstrate that the proposed method is pretty efficient to recognize common gestures with a high accuracy.**

## Keywords

**Human-Computer-Interaction, Gesture Recognition, 3D Point Cloud, Depth Image**

## 1. Introduction

There has been a great emphasis lately on Human-Computer-Interaction (HCI) research to create easy-to-use interfaces by directly employing natural communication and manipulation skills of humans [1]. As an important part of body, naturally, the hand is given more and more attention. Gesture recognition is a key aspect of Human-Computer-Interaction. There are countless researches focus on this advanced topic to create natural user interface and to improve user experiences by using simple and intuitive hand gestures for free-hand controller [2]. How to detect the hands, segment them from the background and recognize the gestures become great challenges. And various methods are proposed to solve those issues.

A hierarchical method of static hand gesture recognition that combines finger detection and histogram of oriented gradient (HOG) features is proposed in [3]. An algorithm based on the spatial pyramid bag of features

**How to cite this paper:** Chong, Y.W., Huang, J.F. and Pan, S.M. (2016) Hand Gesture Recognition Using Appearance Features Based on 3D Point Cloud. *Journal of Software Engineering and Applications*, **9**, 103-111.
http://dx.doi.org/10.4236/jsea.2016.94009

is proposed to describe the hand image in [4]. But both of them are based on RGB image, which means it's difficult to distinguish the hand from complex background. What's more, the intensity of light seriously affects the recognition results. In order to avoid these drawbacks, many scholars select depth image in the research of gesture recognition. Depth information has long been regarded as an essential part of successful gesture recognition [5]. Many researches [6]-[8] extract different features from the depth data, then various classifiers are employed for gesture recognition. These methods all get good effect, but they have to collect a large number of training samples. Based on 3D point cloud data, this paper uses geometric method for extracting the appearance features of gestures and classifying the given gestures. This method can obtain high gesture recognition accuracy without training sample. Compared with the previous methods, it's more succinct and efficient.

## 2. Proposed Gesture Recognition System

The proposed gesture recognition system (shown in **Figure 1**) composed of three parts. In the first part, the 3D point cloud data of the hand region is gotten from depth camera (SwissRanger 4000 depth camera), then after threshold segmentation and gray transformation the 3D point cloud becomes a binary image. Meanwhile, some preprocessing on the grayscale image is necessary. In the second part, some apparent features are extracted. Finally, on the basis of the features extracted in last step the gesture can be recognized.

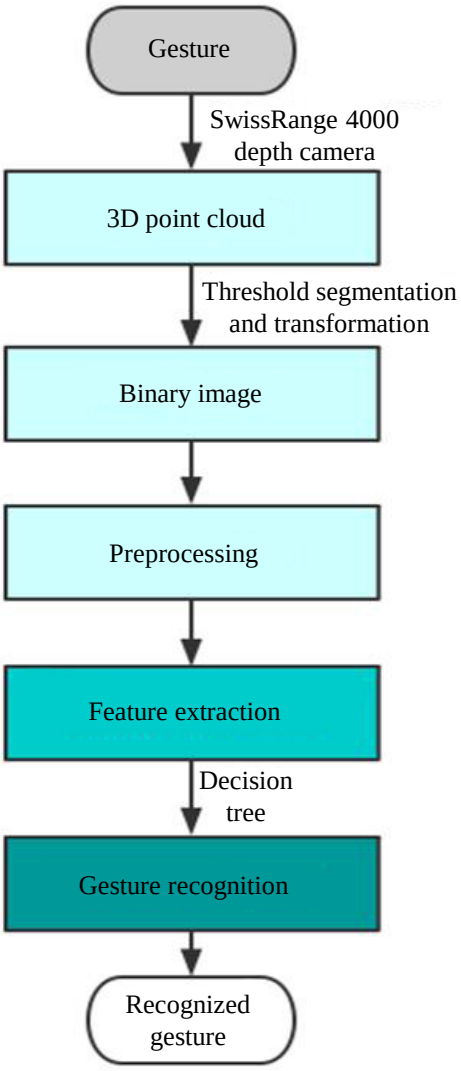


**Figure 1.** Architecture of the proposed gesture recognition system.

## 3. Hand Segmentation

### 3.1. Image Collection

In this paper, the 3D point cloud data of gesture are collected from the SwissRanger 4000 (SR4000) depth camera. The SR4000 cameras are optical imaging systems which provide real time distance data at video frame rates. Based on the Time-of-Flight (ToF) principle, the cameras employ an integrated light source. The emitted light is reflected by objects in the scene and travels back to the camera, where the precise time of arrival is measured independently by each pixel of the image sensor, producing a per-pixel distance measurement. Finally, we can get the three-dimensional coordinates of each point from the camera. A typical 3D point cloud of gesture scene just like shown in **Figure 2**. In this image, the origin of the coordinate system (0, 0, 0) is at the intersection of the optical axis with the front face of the camera, and **Figure 3** shows the camera's output coordinate system.

From **Figure 2** we can notice that the 3D point cloud contains not only hand region but also other region. Obviously, we need to extract the hand region G. G is given by (1):

$$G = \left\{ (x, y, z) \middle| Min_x < x < Max_x, Min_y < y < Max_y, Min_z < z < Max_z \right\} \quad (1)$$

In (1), $Min_x$, $Max_x$, $Min_v$, $Max_v$, $Min_z$ and $Max_z$ are the thresholds of three coordinate directions. We set $Min_x$ = −150 mm, $Max_x$ = 150 mm, $Min_v$ = −150 mm, $Max_v$ = 150 mm, $Min_z$ = 200 mm, $Max_z$ = 500 mm to ensure that the whole hand region is extracted. Next, $G$ is transformed into a binary image.

### 3.2. Image Preprocessing

Due to the nature of the depth sensor, the hand region on the depth map may be have holes and cracks [9], which will seriously affect the accuracy of hand gesture. Usually the binary image always has some noisy. So image preprocessing is necessary, which contains filling the holes and image denoising. In other papers [10]-[12], some inpainting and filtering methods reach a good result. However, the methods are always so complex. We just employ some simple morphological operations (erosion and dilation) in our preprocessing.

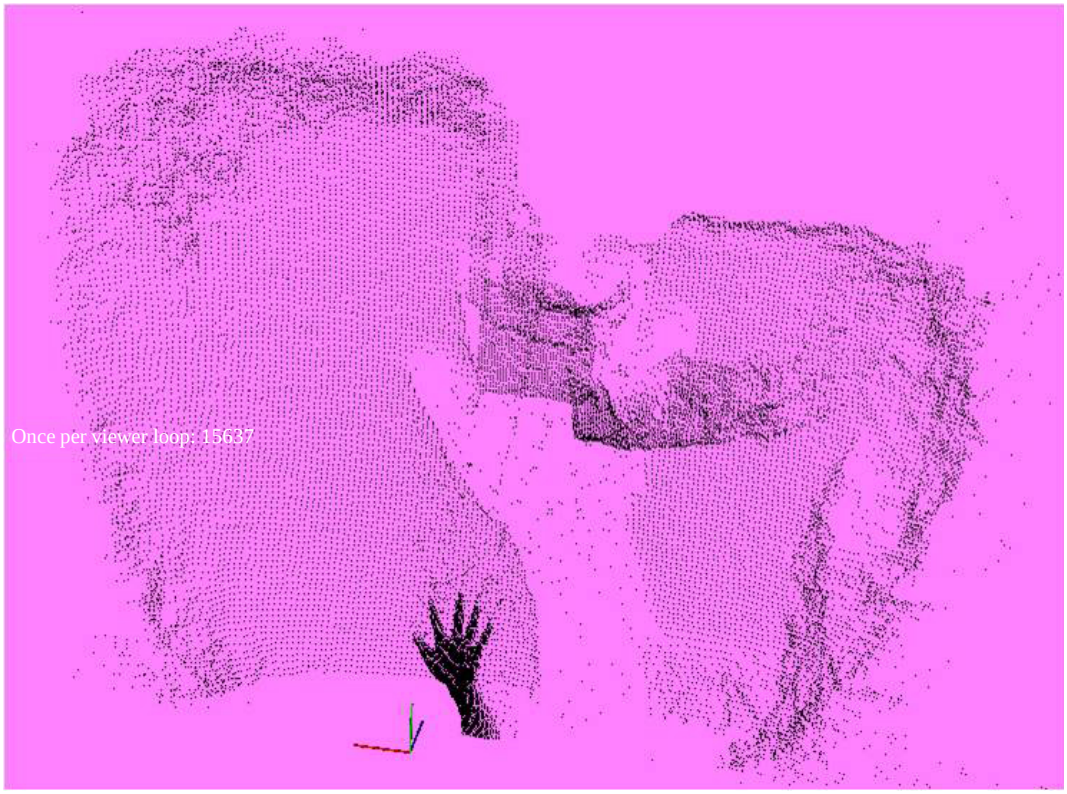


**Figure 2.** 3D point cloud of gesture scene.

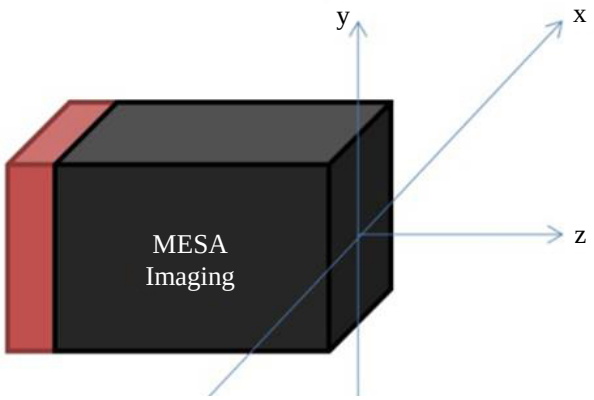


**Figure 3.** (x, y, z) as delivered by the camera is given in this coordinate system

## 4. Extraction of Features

Appearance features are important in gesture recognition. Compared to other methods of feature extraction, appearance features are more intuitional and efficient. As appearance features, the number of stretched fingers and the angles between fingers are used for gesture recognition in [13]. In this paper, we also choose the number of stretched fingers and the angles between fingers as features. Besides, the gesture region's area distribution feature is chosen as an appearance feature, too.

### 4.1. Extraction of the Central Point

Through the erosion operations of mathematical morphology we can locate the central point C. As we all know, the palm is the primary part of gesture. Through continuous erosion operations, the boundary of gesture region is removed over and over again. And the gesture region get smaller and smaller. Eventually, only a point is left, which is just the central point C of the gesture region.

### 4.2. Extraction of Appearance Features

The appearance features used in this paper contain the number of stretched fingers, the angles between fingers and the gesture region's area distribution feature. The following is the main steps of extraction of appearance features.

1) Firstly, the maximum distance value D between the central point and the edge of the gesture region is calculated. Then we $\text{define}_n = n * \frac{D}{10} (n = 1, 2, 3 \cdots 10)$ which represent ten different lengths of radius. Next, choosing C as the centers of rhombuses and $r_n$ as the radiuses, we can draw 10 rhombuses (the innermost one is recorded as the first rhombus and the outermost one is recorded as the tenth rhombus), as shown in **Figure 4** (In order to highlight the effect, the color has been transformed).

2) From **Figure 4** we can notice that every rhombus has a different number of intersections with gesture region. In order to get the number of the stretched fingers N, as a rule thumb, we choose the sixth rhombus to calculate. First of all, in a clockwise direction, we record the points on the sixth rhombus whose color from blue change into red or from red change into blue. We define $K_i$ as the $i$-th (i = 1, 2, 3⋯) point whose color from blue change into red, and $T_i$ as the $i$-th point whose color from red change into blue. Obviously, the number of K or T is just the number of the stretched fingers N.

3) We define $M_i$ as the midpoint of $K_i$ and $T_i$ (i = 1, 2, 3⋯), then each midpoint $M_i$ and the central point C can be connected into a line. And we can calculate each angle between adjacent lines. We use $A_j$ ($j$ = 1, 2, 3⋯$i$-1) to represent these angles.

4) As a rule thumb, the fifth rhombus is chosen as boundary line, so the gesture region is divided into two parts. We define $P_1$ to denote the first part which is inside of the fifth rhombus and define $P_2$ to denote another part which is outside of the fifth rhombus. Then we calculate the ratio of $P_1$ to $P_2$, and we use R to represent this ratio. Naturally, R can be used to describe the gesture region's area distribution feature.

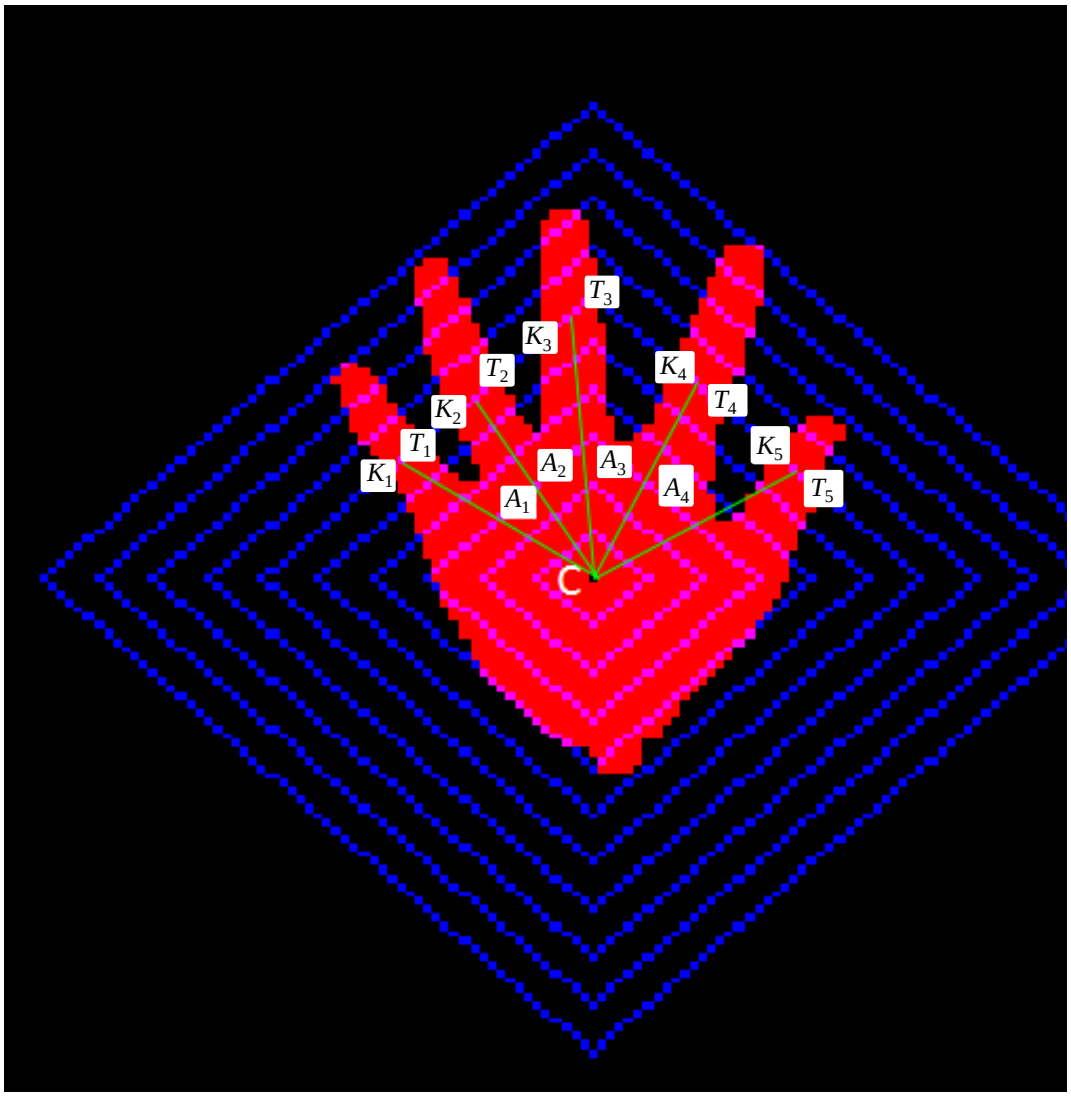


**Figure 4.** Extraction of appearance features.

## 5. Gesture Recognition

Basing on the above appearance features, we construct the decision tree for gesture recognition. In this paper, 9 common gestures (shown in **Figure 5**) are employed for recognition and classification.

Decision tree is a kind of mathematical method to classify the new data by using the decision rules, and the decision rules are get from training samples. The key to construct a good decision tree is to choose the proper logical judgment and attributes.

In this paper, the number of stretched fingers N, the angles between fingers $A_j$ and the ratio of different gesture regions' area R are chosen as branch node of decision tree. First of all, notice that Gesture-9 is the most special gesture, because it doesn't have a stretched finger. So we can choose R as root note of the decision tree to distinguish Gesture-9 and other gestures in the first step. Through some training of samples, we can easily find that Gesture-9's R is always greater than 0.8, and other gestures' R always less than 0.8. So 0.8 is set as a threshold of R. Then, in the rest kinds of gestures, the number of stretched fingers N is an important feature. According to the value of N, Gesture-1, Gesture-4 and Gesture-5 can be uniquely identified. However, if N = 2, the gesture may be Gesture-2 or Gesture-6, if N = 3, the gesture may be Gesture-3 or Gesture-7 or Gesture-8, which is why we need to choose $A_j$ as another appearance feature to distinguish them. Finally, the decision tree we construct is shown as **Figure 6**.

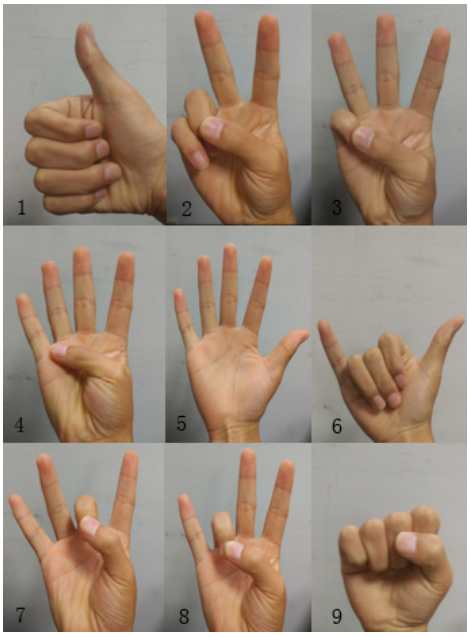


**Figure 5.** 9 common gestures.

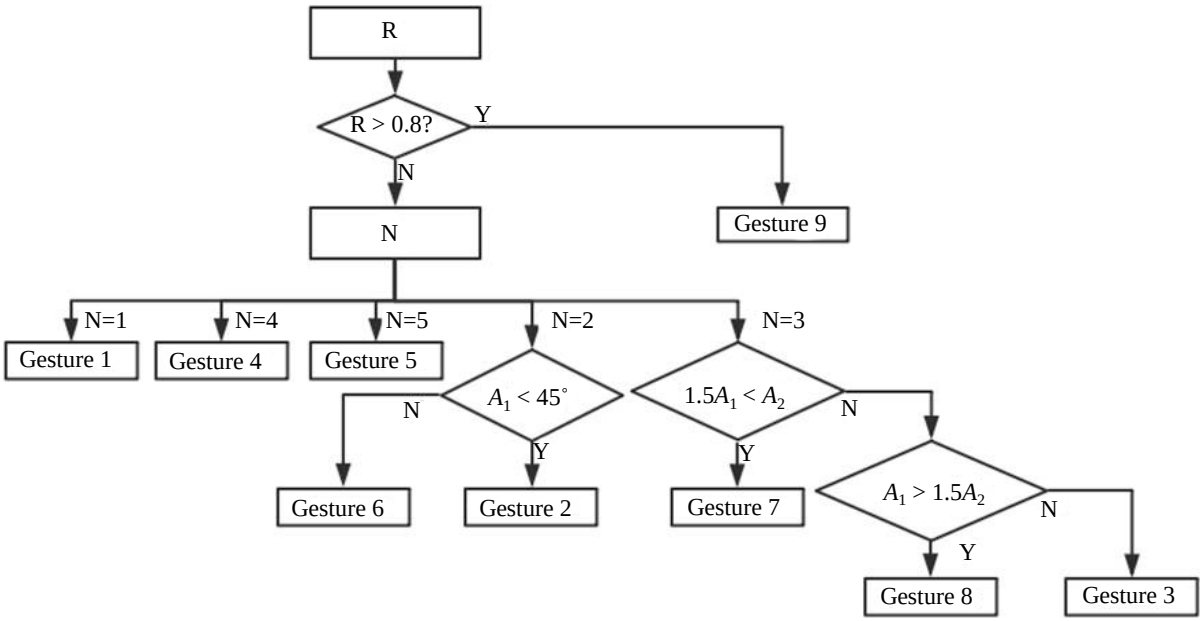


**Figure 6.** The decision tree.

## 6. Experimental Results

In order to validate the method proposed in this paper, we connected the SR4000 depth camera with computer to do a lot of experiments. The experiments were conducted to identify the 9 common gestures. A total of 2700 test samples from 5 people were tested under three different conditions, including under the sunlight (strong light), indoors with the light off (weak light) and indoors with the light on (ordinary light). Each gesture have 300 test samples including different light conditions. The recognition results are shown in **Table 1**, **Table 2** and **Table 3**. In the following tables, the recognition accuracy of each gesture refer to the ratio of the number of correct recognition to the total number of recognition. And the mean accuracy refer to the mean of the recognition accuracy of every gesture.

**Table 1.** The recognition result (strong light).

| | G-1 | G-2 | G-3 | G-4 | G-5 | G-6 | G-7 | G-8 | G-9 |
|---|---|---|---|---|---|---|---|---|---|
| G-1 | 93 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G-2 | 0 | 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G-3 | 0 | 0 | 91 | 1 | 0 | 0 | 5 | 6 | 0 |
| G-4 | 0 | 0 | 0 | 92 | 2 | 0 | 0 | 0 | 0 |
| G-5 | 0 | 0 | 0 | 5 | 97 | 0 | 0 | 0 | 0 |
| G-6 | 1 | 2 | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| G-7 | 0 | 0 | 4 | 0 | 1 | 0 | 89 | 0 | 0 |
| G-8 | 0 | 0 | 5 | 2 | 0 | 0 | 6 | 90 | 0 |
| G-9 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 100 |
| Total | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Recognition Accuracy (%) | 93.0 | 98.0 | 91.0 | 92.0 | 97.0 | 100.0 | 89.0 | 90.0 | 100.0 |
| Mean Accuracy (%) | 94.4 | - | - | - | - | - | - | - | - |

**Table 2.** The recognition result (weak light).

| | G-1 | G-2 | G-3 | G-4 | G-5 | G-6 | G-7 | G-8 | G-9 |
|---|---|---|---|---|---|---|---|---|---|
| G-1 | 94 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G-2 | 1 | 97 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| G-3 | 0 | 1 | 93 | 0 | 0 | 0 | 3 | 3 | 0 |
| G-4 | 0 | 0 | 0 | 93 | 1 | 0 | 0 | 0 | 0 |
| G-5 | 0 | 0 | 0 | 4 | 96 | 0 | 0 | 0 | 0 |
| G-6 | 1 | 2 | 0 | 0 | 0 | 99 | 0 | 0 | 0 |
| G-7 | 0 | 0 | 3 | 2 | 3 | 0 | 90 | 0 | 0 |
| G-8 | 0 | 0 | 4 | 1 | 0 | 0 | 7 | 94 | 0 |
| G-9 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 100 |
| Total | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Recognition Accuracy (%) | 94.0 | 97.0 | 93.0 | 93.0 | 96.0 | 99.0 | 90.0 | 94.0 | 100.0 |
| Mean Accuracy (%) | 95.1 | - | - | - | - | - | - | - | - |

**Table 3.** The recognition result (ordinary light).

| | G-1 | G-2 | G-3 | G-4 | G-5 | G-6 | G-7 | G-8 | G-9 |
|---|---|---|---|---|---|---|---|---|---|
| G-1 | 95 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| G-2 | 3 | 96 | 2 | 0 | 0 | 2 | 0 | 0 | 0 |
| G-3 | 0 | 2 | 90 | 0 | 1 | 0 | 4 | 4 | 0 |
| G-4 | 0 | 0 | 1 | 94 | 1 | 0 | 0 | 0 | 0 |
| G-5 | 0 | 0 | 0 | 3 | 95 | 0 | 0 | 0 | 0 |
| G-6 | 1 | 1 | 0 | 0 | 0 | 98 | 0 | 0 | 0 |
| G-7 | 0 | 0 | 5 | 1 | 3 | 0 | 91 | 0 | 0 |
| G-8 | 0 | 0 | 2 | 2 | 0 | 0 | 5 | 93 | 0 |
| G-9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 99 |
| Total | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Recognition Accuracy (%) | 95.0 | 96.0 | 90.0 | 94.0 | 95.0 | 98.0 | 91.0 | 93.0 | 99.0 |
| Mean Accuracy (%) | 94.6 | | | | | | | | |

**Table 4.** The running time of gesture recognition.

| Gesture | G-1 | G-2 | G-3 | G-4 | G-5 | G-6 | G-7 | G-8 | G-9 |
|---|---|---|---|---|---|---|---|---|---|
| Mean Running Time (ms) | 15.5 | 15.8 | 15.7 | 15.6 | 15.5 | 15.6 | 15.7 | 15.8 | 15.2 |

**Table 5.** Comparative results of the methods in [9], [14] and the proposed method.

| Method | Mean Accuracy | Running Time (s) |
|---|---|---|
| Convex Shape Decomposition Method in [9] | 91.9% | 0.026 |
| Thresholding Decomposition + FEMD in [14] | 90.6% | 0.5004 |
| Near-convex Decomposition + FEMD in [14] | 93.9% | 4.0012 |
| Proposed Method | 94.7% | 0.0156 |

From **Table 1**, **Table 2** and **Table 3** we can notice that the method proposed in this paper has a high recognition accuracy, especially in Gesture-2, Gesture-6 and Gesture-9. The recognition accuracy of Gesture-3, Gesture-7 and Gesture-8 is slightly lower than other gestures, which result from the great difference of the expression of gestures. As a whole, the mean recognition accuracy reach 94.7%, which prove the effectiveness of this method. Meanwhile, another advantage of this method is that the recognition accuracy isn't impacted by light intensity. It can work well in strong light or weak light environment, even in the darkness. In addition, from **Table 4** we can find that the proposed method has extremely short running time. Almost every recognition can be completed within 16ms, in other word, the gesture recognition speed can reach about 60 frames per second, which can completely meet the needs of real-time application. The efficient recognition process ensures that it can be used in real-time situation.

We also compare the proposed system with the previous work. The recognition methods proposed in [14] are geometry-based. Two methods in that paper can't balance accuracy and running time well. And the convex shape decomposition method is employ in [9]. Although the running time get great improvement, the accuracy isn't so satisfying. However, compared with these methods, both the mean accuracy and the running time in our method can reach a pretty good effect. Experiments demonstrate our method is much more efficient for real-time applications, which is shown in **Table 5**.

## 7. Conclusion

Gesture recognition has a wide range of applications in Human-Computer-Interaction. This paper proposes an efficient and succinct method for gesture recognition, which takes advantages of 3D point cloud data. The 3D point cloud data is collected from depth camera, then it is transformed into binary image. Basing on binary image, three different appearance features are extracted, including the number of stretched fingers, the angles between fingers as features and the gesture region's area distribution feature. Finally, the decision tree is constructed for gesture recognition. Extensive experimental results demonstrate accuracy and robustness of the method proposed in this paper. As a result, this method can play a role in the application of real-time gesture recognition.

## Acknowledgements

This paper is funded by National Natural Science Foundation of China (61572372, 41271398), the Foundation Research Funds for the Central Universities (204201kf0242, 204201kf0263), Shanghai Aerospace Science and Technology Innovation Fund Projects (SAST201425).

## References

[1] Ren, Z., Meng, J. and Yuan, J. (2011) Depth Camera Based Hand Gesture Recognition and Its Applications in Human-Computer-Interaction. *Information, Communications and Signal Processing* (*ICICS*), Singapore, 13-16 December 2011, 1-5.

[2] Suarez, J. and Murphy, R.R. (2012) Hand Gesture Recognition with Depth Images: A Review. *RO-MAN*, *IEEE*, Paris, 9-13 September 2012, 411-417.

[3] Liu, S., Liu, Y., Yu, J. and Wang, Z. (2015) Hierarchical Static Hand Gesture Recognition by Combining Finger Detection and HOG Features. *Journal of Image and Graphics*, **20**, 0781-0788.

[4] Yu, S., Cao, J., Li, P., *et al.* (2015) Hand Gesture Recognition Based on The Spatial Pyramid Bag of Features. *CAAI Transactions on Intelligent Systems*, **10**, 429-435.

[5] Janoch, A., Karayev, S., Jia, Y., Barron, J., Fritz, M., Saenko, K. and Darrell, T. (2013) A Category-Level 3D Object Dataset: Putting the Kinect to Work. In: Fossati, A., Gall, J., Grabner, H., Ren, X.F. and Konolige, K., Eds., *Consumer Depth Cameras for Computer Vision*, Springer London, London, 141-165. http://dx.doi.org/10.1007/978-1-4471-4640-7_8

[6] Dominio, F., Donadeo, M. and Zanuttigh, P. (2014) Combining Multiple Depth-Based Descriptors ForHand Gesture Recognition. In: Borgefors, G., Sanniti di Baja, G. and Sarkar, S., Eds., *Pattern Recognition Letters*, Elsevier, Amsterdam, 101-111. http://dx.doi.org/10.1016/j.patrec.2013.10.010

[7] Nguyen, L.T., Thanh, C.D., Ba, T.N., Viet, C.T. and Thanh, H.L. (2013) Contour Based Hand Gesture Recognition Using Depth Data. *Advanced Science and Technology Letters*, **29**, 60-65.

[8] Konda, K.R., Königs, A., Schulz, H. and Schulz, D. (2012) Real Time Interaction with Mobile Robots Using Hand Gestures. *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, Boston, 5-8 March 2012, 177-178.

[9] Qin, S., Zhu, X., Yang, Y. and Jiang, Y. (2014) Real-Time Hand Gesture Recognition from Depth Images Using Convex Shape Decomposition Method. *Journal of Signal Processing Systems*, **74**, 47-58. http://dx.doi.org/10.1007/s11265-013-0778-7

[10] Daribo, I. and Saito, H. (2011) A Novel Inpainting-based Layered Depth Video for 3dtv. *IEEE Transactions on Broadcasting*, **57**, 533-541. http://dx.doi.org/10.1109/TBC.2011.2125110

[11] Telea, A. (2004) AnImage Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools*, **9**, 23-34. http://dx.doi.org/10.1080/10867651.2004.10487596

[12] Kopf, J., Cohen, M.F., Lischinski, D. and Uyttendaele, M. (2007) Joint Bilateral Upsampling. *ACM Transactions on Graphics*, **26**, 1-8. http://dx.doi.org/10.1145/1275808.1276497

[13] Cao, C., Li, R. and Zhao, L. (2012) Hand Posture Recognition Method Based on Depth Image Technology. *Computer Engineering*, **38**, 16-18.

[14] Ren, Z., Yuan, J. and Zhang, Z. (2011) Robust Hand Gesture Recognition Based on Finger-Earth Mover's Distance with a Commodity Depth Camera. *ACM International Conference on Multimedia*, Scottsdale, 28 November-1 December 2011, 1093-1096. http://dx.doi.org/10.1145/2072298.2071946