IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume 7,Issue 4,Dec2019

PERFORMANCE OF HAND GESTURE RECOGNITION BASED ON HIGH-LEVEL FEATURES

[1]Dr. Ch Venkateswararao, [2]A Narasimha reddy, [3]K Srinivasulu, [4]V Chinnarao

ABSTRACT

Gesture recognition plays an important role in human-computer interaction. However, most existing methods are complex and time-consuming, which limit the use of gesture recognition in real-time environments. In this paper, we propose a static gesture recognition system that combines depth information and skeleton data to classify gestures. Through feature fusion, hand digit gestures of 0-9 can be recognized accurately and efficiently. According to the experimental results, the proposed gesture recognition system is effective and robust, which is invariant to complex background, illumination changes, reversal, structural distortion, rotation etc. We have tested the system both online and offline which proved that our system is satisfactory to real-time requirements, and therefore it can be applied to gesture recognition in real-world human-computer interaction systems.

Keywords: Gesture recognition; Kinect; human-computer interaction; data fusion; real-time

1. Introduction

With the extensive application of computer, human-computer interaction has become an important part in people's daily life with an ultimate goal to realize the natural communication between human and machine. The hand is the most flexible connector between human and the real world, so does the connection between human and the computer based on physical sensory data. Gestures are composed of different posture and motion. However, hand is an articulated object with more than 20 degrees of freedom,[1] and occlusions occur in its projection results. In addition, hand movements are usually too fast and complex compared to the current speed of image processing methods. As a result, real-time gesture estimation is still

[1]Professor,Dept. of ECE, RISE Krishna Sai Gandhi Group of Institutions, Ongole
[2,3,4]Asst Professor, Dept. of EEE, RISE Krishna Sai Gandhi Group of Institutions, Ongole

a challenging research topic involving multidisciplinary work such as pattern recognition, image processing, computer vision, artificial intelligence and machine learning. Gesture recognition, which analyzes the gesture meanings to complete a friendly and easy interactive experience, has played a pivotal role in many fields, such as virtual reality,[2] augmented reality,[3] sign language recognition,[4] and health care.[5] Different from the traditional mouse and keyboard input, human-computer interaction based on gestures is more natural, direct and with less limit.Although gesture recognition is a natural and intuitive process in human-computer interaction, it is possible to produce a large number of different combinations. Besides, the high flexibility of human hands increases the recognition difficulty mainly from the following two aspects. One obstacle is the hand detection and tracking. Data glove or mark-based methods provide a viable solution,[6] but wearing extra equipment is inconvenient and uncomfortable for users. Vision-based gesture recognition, affected by the constraints of traditional cameras, is more natural with low-price cameras. However, hand segmentation cannot well overcome the impact of external factors, resulting in that hand gesture recognition methods are sensitive to illumination conditions and complex background. Besides, the calculation process is complex and the real-time performance is poor. The other obstacle is that simple features often lead to ambiguities and low accuracy, while complex features require more processing time due to the complexity of the hand posture. Therefore, it is very important to choose suitable features for recognizing different hand gestures.

The development of gesture recognition is driven by a recent evolution, Microsoft Kinect, which has been successfully applied to a variety of computer vision tasks, such as face recognition,[7] [8] scene reconstruction,[9] and gesture recognition.[10] This device is equipped with an RGB camera, a depth sensor and a multi-array microphone. Through Kinect, we can track the user's body movement and obtain color and depth information, and thus enjoy a better experience in human-computer interaction. Fig.1 shows Kinect V2 for Xbox One. There are many methods for gesture recognition,[11] [12] [13] such as depth image-based, skeleton data-based and shape matching. Unfortunately, the classic shape recognition methods based on shape matching[14] and skeleton matching[15] cannot reliably identify shape contours with severe deformation. Depth images provided by Kinect are not affected by complex background and can provide the basis for accurate gesture segmentation. To this end, we use Kinect as a data acquisition device in our gesture recognition system.

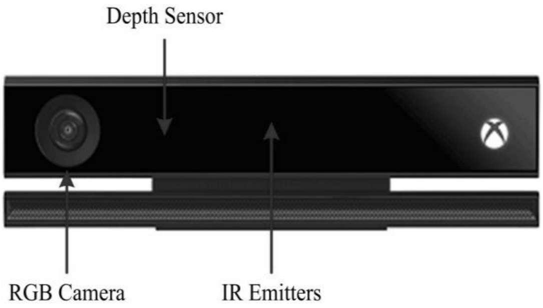


Fig.1. Kinect V2 for Xbox One.

In this paper, we propose a static gesture recognition system that combines depth information and skeleton data to classify gestures. Through combining the features such as fingertips, vector angle and SIFT keypoints, our system can recognize digit gestures of 0-9. Generally, the features used for gesture recognition can be divided into low-level and high-level categories. As a high-level feature, fingertips contain sufficient information that can be used to judge whether the fingers are stretched or bent and then make some simple identification of gestures. So we choose the fingertips as the main feature for gesture recognition. For some gestures that cannot be recognized by a single fingertip feature, there is a need to select some sub-features for secondary classification, in which we choose the angle between vectors and SIFT keypoints as the secondary classification features according to the shape property of gestures. Through the combination of these features, we can successfully classify 10 gestures.

Fig.2 shows the hand shapes to the corresponding digit gestures. The specific gesture recognition process is as follows: 1) depth information and skeleton data are obtained through the Kinect; 2) a double-threshold segmentation algorithm is adopted to extract the hand mask and smooth it; 3) themask contour is extracted and the shape descriptor (fingertip) of the connected area is calculated; 4) features including fingertips, angle between vectors and SIFT keypoints are fused to classify the gestures, and 5)the gestures are quickly and accurately identified, and then displayed on the screen.

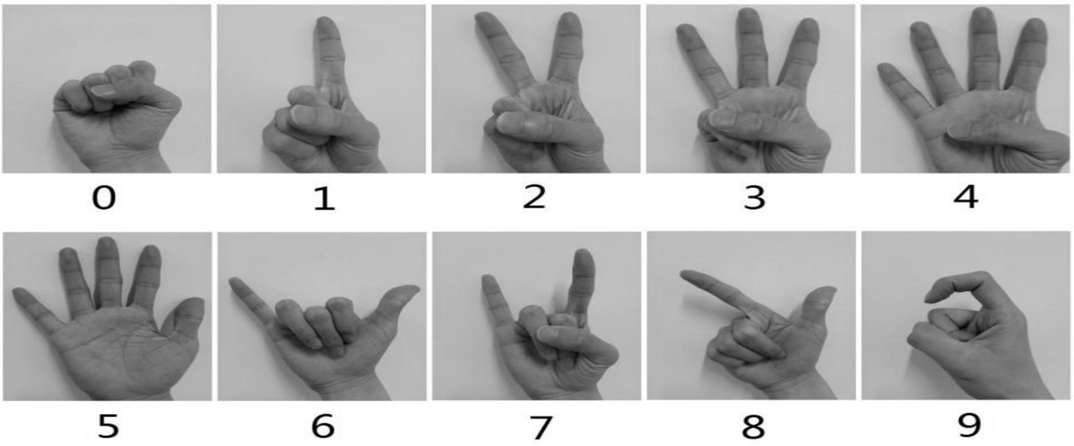


Fig.2. The hand shapes corresponding to the digit gestures from 0 to 9.

In our proposed gesture recognition system, we contribute to overcoming the following difficulties for hand gesture recognition:

(i) Extract stable high-level features to accurately classify gestures;

(ii) Track and recognize gestures very efficiently to satisfy the real-time requirements; and

(iii) The system is robust to rotation, flip, illumination changes, background confusion and distortions. The remainder of

(iv) this paper is organized as follows. Related work is discussed in Section 2. Section 3 describes our proposed gesture recognition system. In Section 4, the experimental results are described. We summarize the paper in Section 5.

2. Related Work

In general, a vision-based gesture recognition system consists of three parts: hand segmentation, feature extraction, and gesture classification, as shown in Fig.3. Hand segmentation attempts to extract the hand boundary in an image; the purpose of feature extraction is to find identification information in the image for the subsequent gesture classification which determines the gesture that best matches the extracted feature of the input image.

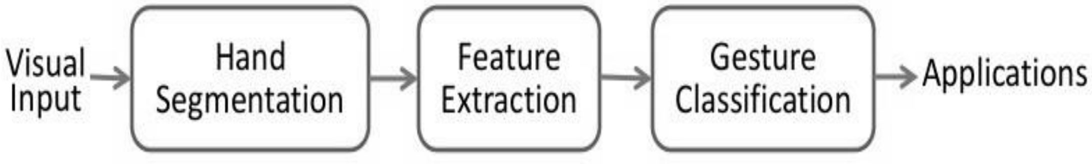


Fig.3. The pipeline of vision-based hand gesture recognition

2.1. Hand segmentation

The purpose of hand segmentation is to detect the hand. According to the data source to be distinguished, existing hand segmentation algorithms can be divided into two categories: RGB images-based methods and depth images-based algorithms, which are introduced as follows:

(i) RGB image-based methods: they use a single or multiple cameras to collect gesture information, which is processed to segment hand. There are mainly two hand segmentation methods based on RGB images. One is based on the apparent characteristics referring to skin color, texture, fingertips, contours, hand type and size, etc. Skin color is the most commonly used characteristics usually extracted and modeled in YUV and HSV color spaces for obtaining the segmented hand. Han et al. proposed an algorithm for skin

(ii) color segmentation based on adaptive threshold

But this algorithm is easily affected by the external environment, such as illumination changes and other skin color objects. The other is adding the constraints on background artificially. For example, the use of black and white walls, dark clothing and other simplified background. In this way, researchers can simplify the division of the hand area and the background area. Farid et al. proposed a gesture recognition system based on data gloves,[17] which uses proprietary techniques to convert hand and finger movements into real-time digital joint data. Artificially adding the constraints on background can solve the interference of skin color, but the environment is more demanding, which reduces the natural expression of hand.

(iii) Depth images-based algorithms: Using depth images can achieve more accurate and reliable segmented results, and is able to resist the interference of

(iv) complex background and illumination changes. In recent years, depth cameras (such as Kinect and Leap, etc.) have brought new possibilities to gesture recognition. Kinect that can provide high-quality depth images was used as one of the most popular acquisition device, and the Finger-Earth Mover's Distance (FEMD) was used as the standard for template matching.[18] An algorithm based on stochastic decision forest was proposed to well solve the static and dynamic hand gesture recognition problems.[19] A method based on random regression forest was proposed to extract the hand skeleton in real time.[20] The algorithm is robust to self-occlusion and low resolution of depth camera.

2.2. Feature Extraction

The quality of the features extracted from the input image plays an important role in gesture recognition. Generally speaking, features can be divided into low-level features and high-level features, as shown in Fig.4. Low-level features (such as edges, contours and skin color) can be simply extracted and have been widely used for gesture recognition. But only relying on these low-level features cannot guarantee a satisfied recognition result. High-level features (such as fingers, fingertips and palm parts) provide a more compact representation of the raw visual inputs, and thus make gesture recognition easier. However, it is still challenging to extract these high-level features robustly. In summary, we need to consider the specific requirements and environment to combine the low-level features with high-level features for gesture recognition.

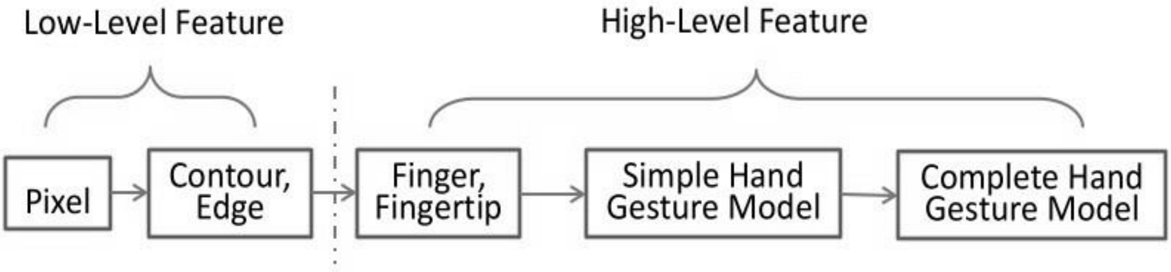


Fig.4. The generally used low-level features and high-level features in gesture recognition.

As skin color is invariant to scaling and rotation, it is widely used for hand location detection.[21] The edge as another common low-level feature is usually used for gesture recognition based on template matching.[22] In many gesture recognition systems,[23] edges are used jointly with other features (such as skin color and contours) since the feature fusion can improve the accuracy and robustness of the algorithm.[24]

Fingertip position, finger location and gesture orientation are related to the hand structure, which can be easily matched to the raw visual inputs. Fingertip detection and finger type determination are studied with a model-based method.[25] Besides, a finger detection method using grayscale morphology and blob analysis was described,[26] which can be used to detect bending fingers. In some cases,high-level features can also be extracted from low-level features. For example, high-level hand features were extracted by analyzing the hand contour.[27] Nevertheless, high-level features are not easy to be extracted accurately.

2.3. Gesture Classification

Vision-based gesture recognition has been extensively studied, but it is still a challenging task to classify gestures from complex backgrounds, due to the high degree of flexibility and self-occlusion. Following describes the current situation of static gesture recognition. Static gesture recognition methods can be divided into the following categories:

(i) Edge feature matching-based: Gesture recognition based on edge feature matching is achieved by calculating the relationship of feature sets between the test samples and the originally saved samples, and then finding the best matches.[28] The edge feature extraction is relatively simple and adaptable for complex background and illumination changes. However, the data-based matching algorithm is quite complex and time consuming for identifying different gestures since it needs to prepare a large number of templates.

(ii) Haar-like feature-based: Adaboost recognition based on Haar-like features can be used for hand detection and simple gesture recognition.[29] [30] This method can identify specific gestures in real time under complex background, but it is

(iii) not robust to severe posture deformation. Nevertheless, hand gestures have a variety of shape changes, and therefore this method can only be applied in predefined static gesture recognition.

(iv) External contours-based: In general, different gestures have different external contours. The curvature of the external contour is different at different positions of the hand (i.e., the curvature of the fingertips is relatively large). By analyzing the curvature scale space,[27] researchers extracted the corresponding features and achieved the purpose of gesture recognition.

(v) Finger characteristics-based: Finger is the most widely used feature in gesture recognition, because the location and states of the finger represent the most intuitive characteristics of different gestures. Just combining finger location, finger state and palm position can directly identify simple gestures. The finger was modeled as a cylindrical object according to the parallel edge feature of the finger, and then the finger was extracted from the prominent edge.[31] After the hand was segmented, a radial projection algorithm was used to identify the gesture. Xavier et al. presented an algorithm for obtaining the fingertip position from depth maps by using a feature called radial distribution that not only describes the gesture from the overall,[32] but also locates the fingertips from local information.

2.4. Research Progress

During the last few years, depth cameras (such as Kinect and Leap, etc.) have brought new possibilities to gesture recognition. For example, Tang used both RGB and depth data obtained from Kinect for detecting hand gestures.[33] The approach can recognize ‘grasp’ and ‘drop’ gestures with over 90% accuracy. Although these two types of gestures are dynamic which increases the recognition difficulty, the two-type recognition task is relatively simpler compared with that in our proposed recognition system. Moreover, Tang built three independent features to test the recognition performance, including Radial Histogram, Raw Pixel Estimates and Modified SURF. While in our recognition system the features are combined, this is a new attempt. Kinect, one of the most widely used depth devices, provides high-quality depth images to effectively solve the problems of complex background and lighting changes. Due to the superiority of depth images, gesture recognition using the information from depth images becomes a new trend.The usage of depth information brings great help to hand tracking and hand segmentation. Satisfactory hand segmentation result provides a solid foundation for feature extraction. However, a simple feature cannot meet the requirements for more complex gesture recognition, and thus many manually extracted features have been developed for gesture recognition. Zhang et al. proposed Histogram of 3D Facets (H3DF) by utilizing rich 3D shape information that can effectively represent both 3D shapes and structures of various depth maps.[35] By combining H3DF with an N-gram model and dynamic programming, this gesture recognition system achieved good performance of 99% accuracy on the NTU hand digits and the ASL finger spelling dataset. Zhou et al. proposed a high-level feature extraction method by extracting the fingers from the parallel edge characteristic of hand and then utilizing a radial projection algorithm to locate each finger for recognizing the gestures.[31] Fang et al. proposed an electromyography (EMG) feature named differential root mean square (DRMS),[36] which takes the relationship between neighboring EMG channels into account, and the proposed system can recognize four different hand motions. Pisharady et al. proposed a novel approach for hand posture and face recognition based on fuzzy-rough sets,[37] and features of the image are extracted using the computational model of the ventral stream of visual cortex. The proposed algorithm is tested using two hand posture and three face datasets which provided good classification accuracy. These methods have achieved good recognition results, but designing these features is complicated and time consuming. By comparison, the features we extracted are relatively simple since we only combine fingertips, angle between vectors and SIFT keypoints. In this way, wenot only save the time but also improve the recognition efficiency.

3. Methods

Since the hand has more than 20 degrees of freedom and sometimes moves with the arm, the segmented hand area often contains part of the arm, which brings a considerable difficulty to gesture recognition. It is also worth noting that the fingers often entail self-occlusion, especially the bending fingers, increasing the difficulty of feature extraction.

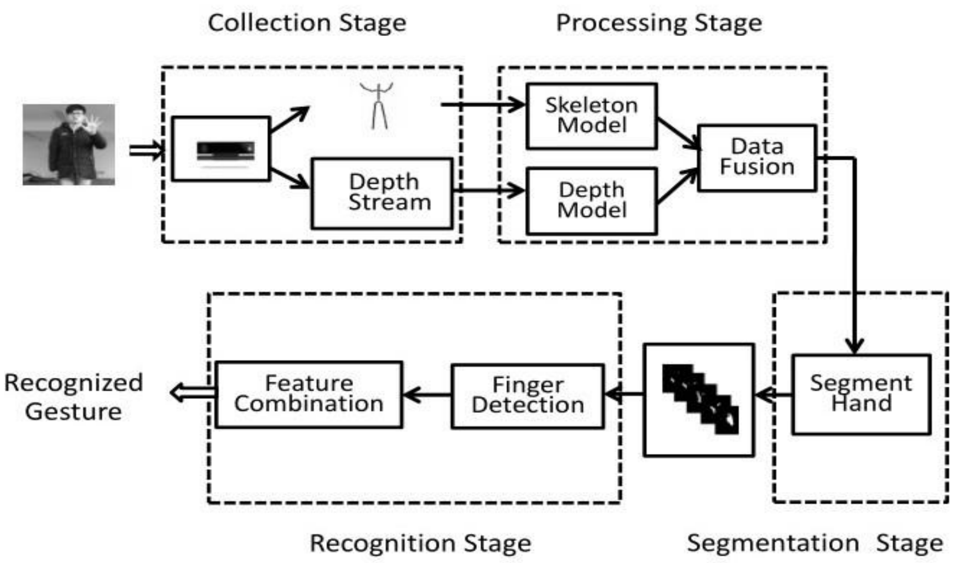


Fig.5. The general framework of our hand gesture recognition system

In order to solve the above-mentioned problems of hand segmentation and feature extraction, we propose a gesture recognition system based on data fusion. We use Kinect as the acquisition device, and choose the fingertip as the main feature. The gesture recognition process consists of the following steps: 1) Kinect V2 is used to acquire depth information and skeleton data; 2) the mask is obtained by hand segmentation based on the fusion of depth information and skeleton data; 3) the mask contour is extracted and the shape descriptor (fingertip) of the contour area is calculated; and 4) the features including the fingertips, angle between vectors and SIFT keypoints are combined to accurately and efficiently classify the gestures. The general framework of the system is given in Fig.5, and each step isdescribed in detail in the following subsections.

3.1.Data Acquisition

Kinect is a human-computer interaction device developed for XBOX360 by Microsoft. Kinect sensor uses deep sensing technology (built-in color camera, infrared transmitter and a microphone array) to perceive human position, action and sound since it provides color image streams, depth image streams, skeleton image streams and infrared image streams in real time. Microsoft has released the Kinect for Window SDK, and we can easily access the data stream through the Kinect API. Fig.6 shows sample data streams obtained by Kinect.

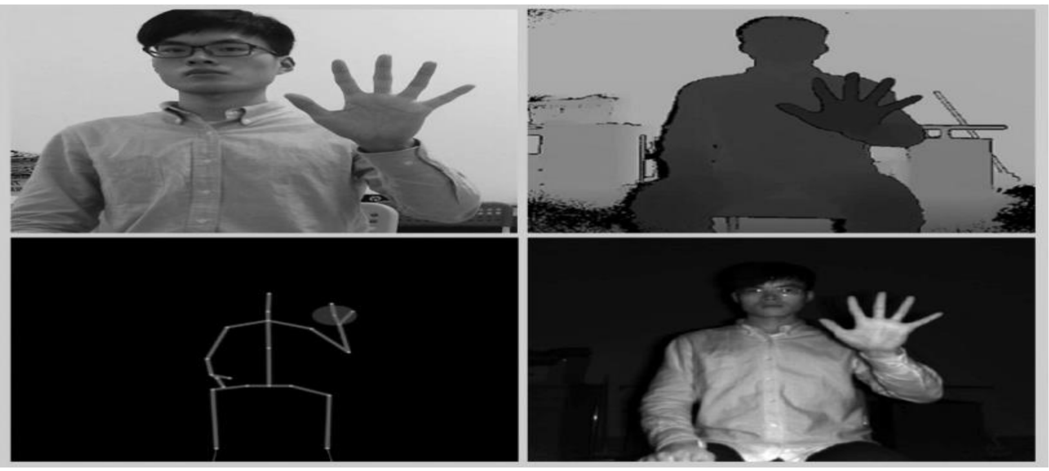


Fig.6. From left to right and top to bottom, there are a color image, a depth image, a skeleton image and an infrared image.

3.1.Data Fusion and Hand Segmentation

In our system, we only use depth images with the size of 512 * 425 and 25 joints of skeleton data. We use skeleton data to locate the hand joint and then map the hand joint position to the depth image. Using only the depth information and skeleton data can segment the hand, so there is no need to adopt other data streams. The fusion results are shown in Fig.

7.

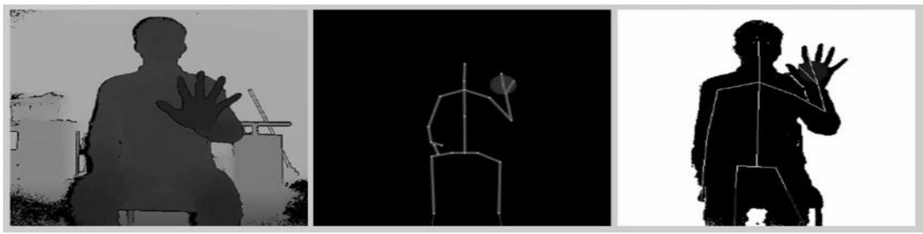


Fig.7. Fusion of depth information and skeleton data.

The next step is to use the depth value of the depth image to separate the hand from the fused image. We use the double-threshold segmentation method to separate the hand. The depth threshold is chosen according to the depth value of the central palm, which determines the approximate position of the hand. Then, we set a positive and a negative deviation of 50 mm about the depth value. That is to say, objects in this 100 mm interval are considered as hand.

The specific steps are as follows:

(1) After acquiring depth images from Kinect V2, we obtain hand skeleton points including wrist and central palm. These two points are mapped onto depth images and their depth values are saved for future use;

(2) As every skeleton point in 3D space is associated with a position and an orientation, we can obtain the position of the central palm and wrist in real time;

(3) A rectangle region is selected (for example with size 200*200) around the central palm in depth images. Initial segmentation is conducted based on the depth value of the central palm. Because the right hand more conforms to the habit of human-computer interaction, we choose the right hand as the identification target;

(4) The rectangle is traversed to find the pixel that matches the depth value. And the pixel that matches the depth value is set to 1, otherwise 0;

(5) The segmented hand area sometimes contains part of the arm, which may not ensure a precise division. To this end, we use the wrist skeleton points in the depth of the two-dimensional coordinates for precise segmentation;

(6) The binary image is smoothed using median filter and we set 5 as the aperture linear size.

Fig.8 shows the depth images of gesture 2 and gesture 8, and their corresponding hand masks. Fig.9 shows the segmentation results for digit gestures from 0 to 9.

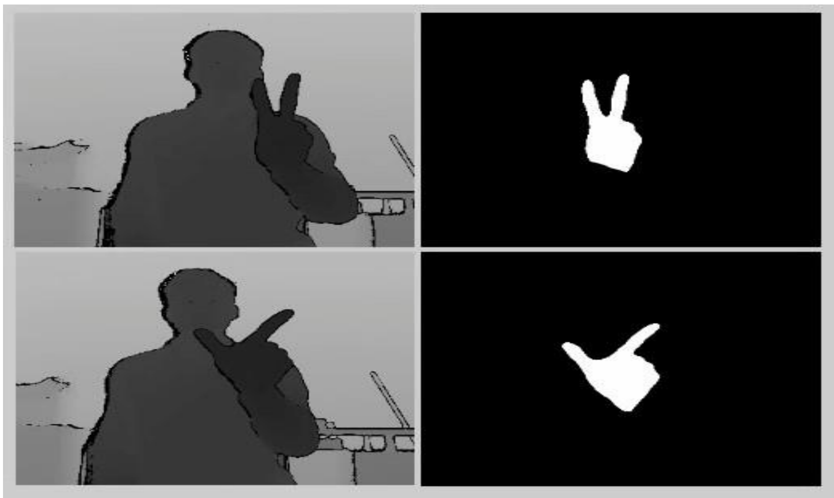


Fig.8. Depth images and their corresponding segmented hand masks.

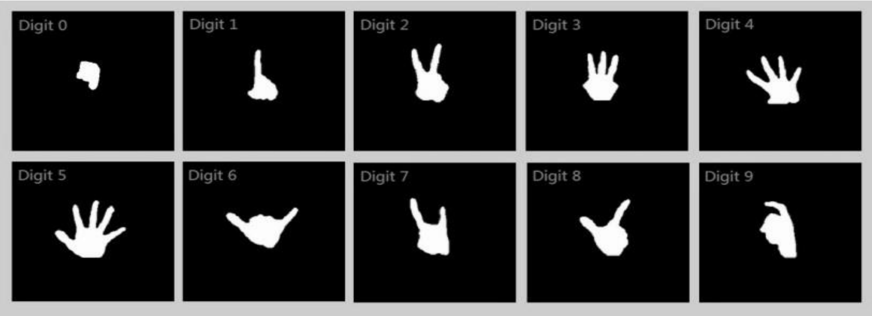


Fig.9. Segmentation results of digit gestures from 0 to 9.

3.1. Feature Extraction

3.1.1. Finger Detection

The mask is a low-level feature which only contains a small amount of information, and thus is difficult to classify multiple gestures directly according to the mask shape. One of the most intuitive features that can be extracted from the mask is the contour, which is made up of a series of points. Fig.10 shows the external contours of gestures. By analyzing these points, we can extract some high-level features such as fingertips, convex and concave. The features containing more meaningful information indicate easier identification process and higher identification accuracy. As a high-level feature, fingertips contain sufficient information that can be used to judge whether the fingers are stretched or bent and then identify some simple gestures. Therefore, we choose the fingertips as the main feature for gesture recognition

.

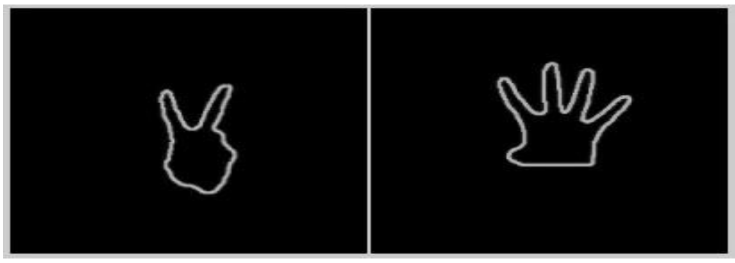


Fig.10. Contours of gesture 2 and 4.

We use the distance between gravity centers to detect the fingertips and obtain the shape descriptor (fingertip) of the contour area. Fig.11 shows the fingertips of the gestures 3 and 4, where the red

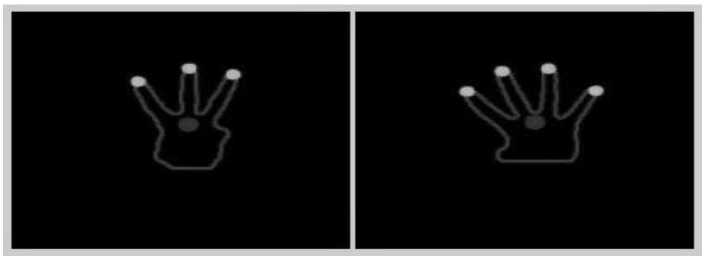


Fig.11. Fingertips and gravity center of gesture 3 and 4, respectively

dot is the gravity center of the hand and the green dots are detected fingertips.

All the points of the hand external contour are traversed by the distance to gravity center in a clockwise or counterclockwise direction, and the peak point of the contour is regarded as the candidate fingertip points. According to the finger characteristics under normal circumstances, the finger's y coordinate should be higher thanthe coordinate of palm center. When traversing the peripheral contour and comparing the candidate point coordinates with the palm center coordinates, the coordinates of candidate points that are lower than that of the palm center will be removed.

3.1.2. Angle Calculation

For digit gestures of 0-9, we cannot accurately recognize each gesture simply by the number of fingertips since two images with the same number of fingertips may indicate two different gestures. As shown in Fig. 12, the number of fingertips among those four gestures is two, but it refers to 2, 6, 7, and 8, respectively. Therefore, a secondary classification is needed for accurate identification. We found gestures can be classified by the angle characteristics: the fingertip and the gravity center can form a vector, and the angles between these vectors are different. Assuming that the coordinates of the gravity center (x, y), the coordinates of two fingertips (x1, y2) and (x2, y2), and the angle of the two vectors θ, the angle's cosine of the two vectors can be computed by Eq. (1). After setting the corresponding angle threshold of these gestures, the actual value is compared with the threshold for accurate gesture recognitio n.

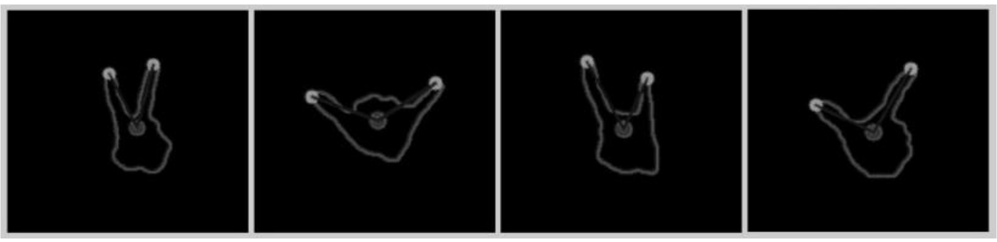


Fig.12. Recognition results of gestures 2, 6, 7 and 8.

3.3.3 SIFT Keypoints

For the gestures 1 and 9 that have the same amount of fingertips and cannot be classified by the angle, a secondary classification is also needed. Because the difference between these two gestures is mainly on bending characteristics, the use of global features such as circumference, area and other features cannot achieve a good recognition result. As a local feature, SIFT has favorable uniqueness and adequate information,[38] which has a strong invariance to most transformations or disturbances like rotation, scaling, different angles of view, and noise. For this reason, we use SIFT keypoints as the feature for secondary classification.

The SIFT keypoints are obtained by the following steps: (i) set up the scale space of difference of Gaussians (DoG) and detect the extreme points in adjacent scales of the DoG scale space; and (ii) determine the location and scale of the key points accurately through the three-dimensional quadratic fitting function, meanwhile excluding the points with low contrast and unstable edge responses

.

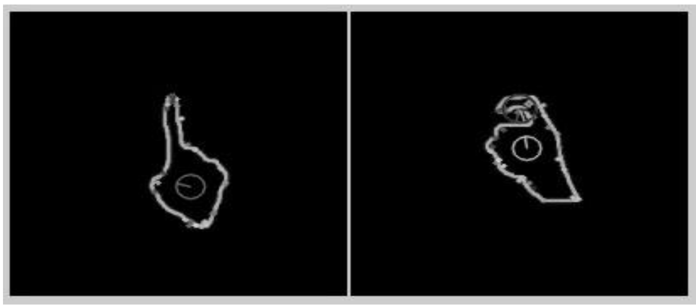


Fig.13. SIFT keypoints of gesture 1 and 9.

3.4 Hand Gesture Recognition

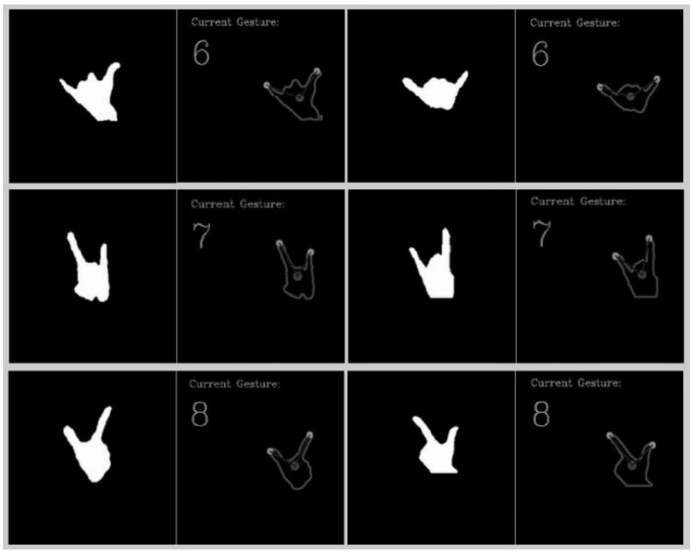
In summary, we choose fingertips as the main feature for gesture recognition, and we also introduce the angle characteristics and SIFT for secondary classification.The brief procedure for gesture recognition is as follows. The number of fingertips is obtained through fingertip detection. When the number is 0, 3, 4, 5, it is the corresponding digit; when the number is 1, the gesture is determined to be 1 or 9 by detecting the amount of SIFT keypoints for secondary classification; when the number is 2, the gesture is identified to be 2, 6, 7 or 8 by calculating the angle characteristics for secondary classification.

4. Experimental Results

In this section, we tested the proposed gesture recognition system to verify its efficiency androbustness. All experiments were conducted on a personal computer with an Intel CORE i5 2.5 GHz processor (4GB RAM). In order to verify the validity of the proposed feature extraction algorithm, we designed gesture recognition systems both online and offline.

4.1. Online System

When recognizing the gesture online, the user can sit on a chair or stand in front of Kinect V2, move forward or backward (left or right), making the identification task more challenging. Fig. 14 shows the recognition results for digit gestures of 0-9.



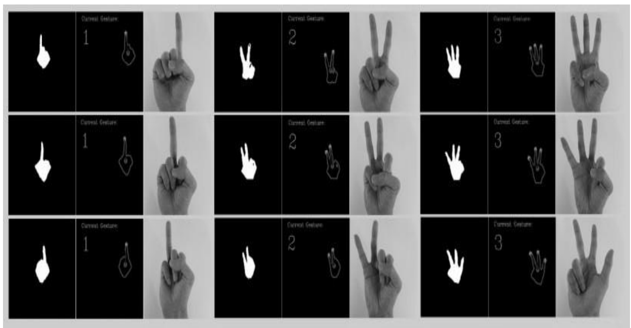


Fig.15. Invariance to reversal

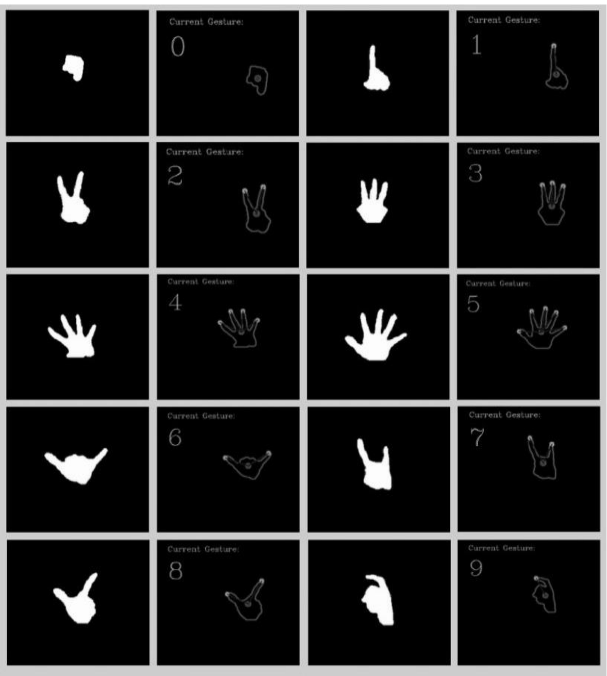


Fig.14. From left to right and top to bottom, recognition results for digit gestures of 0-9.

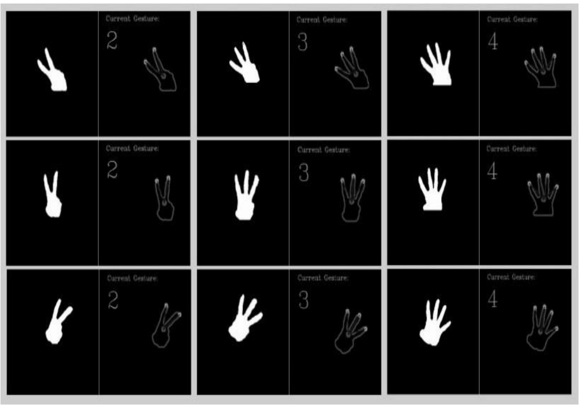


Fig.17. Invariance to rotation.

Beside,we verify the robustness of the proposed gesture recognition system. Because the depth images we used are based on the Time of Flight (TOF) principle, the gesture recognition system has the invariance to complex backgrounds and illumination changes. Not only that, since the gestures 0-9 are identified by the number of fingertips and the combination of fingertip, angle and SIFT features, the proposed gesture recognition system has a good invariance to reversal, structural distortion and rotation, as shown in Fig. 15, 16 and 17.

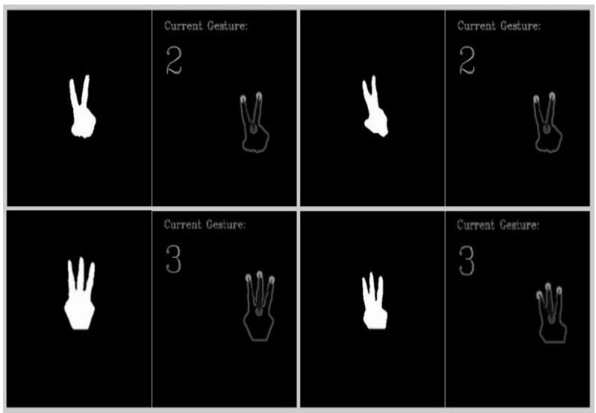


Fig.16. Invariance to structural distortion.

For gestures 1, 2 and 3, our system can identify different expression types to adapt the habits of different people. Fig.18. shows the diversity of gestures.

Fig.18. Expression diversity of gestures.

In order to test the real-time performance of the online gesture recognition system, for each gesture, we perform 20 times during the test, and then calculate the average time for recognizing each gesture. As shown in *Table 1*, except for the gesture 1 and 9 that take a little longer to extract SIFT features, the recognition time of the other gestures falls into the range of 50-60ms, which is reduced by 27.5% compared with the current advanced gesture recognition algorithm.[18] This indicates that our algorithm meets the requirements of real-time gesture recognition in real-world applications.

Table 1. Time cost of each gesture tested in the online system.

Gesture	0	1	2	3	4	5	6	7	8	9
Time (ms)	51.7	404.9	54.6	53.9	55.9	57.1	53.5	52.1	56.5	408.4

4.2. Offline System

In order to verify the effectiveness of the feature extraction algorithm and to ensure the integrity of the gesture recognition research, we designed an offline gesture recognition system. We use the RGB images as the visual input and the skin color segmentation algorithm to separate hands, and the remaining steps are as the same as in the online gesture recognition system. In the stage of gesture segmentation, the RGB color space is converted to the HSV color space, and the corresponding threshold is set to extract the hand mask. We use a traditional camera (imx258-sunny) to collect the gesture images of 10 volunteers, resulting in 100 images in total. Since we use skin color to segment the hand, the background of the input RGB images requires relatively simple and non-skin like objects, and artificially adding the constraints on background can achieve a better segmentation result. *Table 2* shows the average recognition time and accuracy of offline system. We find that the recognition time is shorter compared with that in the online system, since the image pre-processing takes a little more time in the online system. Besides, the recognition accuracy is relatively higher except for gesture 0 and 1. Because we use the distance of gravity centers to detect the fingertip, and the distance between the fingertip and the palm center is similar for gesture 0 and 9, it often makes a mistake that gesture 0 was recognized as gesture 9. As for gesture 1, since the segmented image contains little noise that will increase the number of SIFT keypoints, gesture 1 may be mistaken for gesture 9. In general, the feature extraction algorithm we proposed is effective to meet the requirements of gesture recognition in real-world human-computer interaction systems.

Table 2. Time cost and accuracy of each gesture tested in the offline system.

Gesture	0	1	2	3	4	5	6	7	8	9
Time (ms)	26.3	223.7	28.7	32.1	37.6	35.5	28.3	31.7	35.9	231.7
Accuracy(%)	48	59	98	96	94	90	95	93	96	83

5.Conclusion and Future Work

In this paper, we propose a high-level gesture recognition system that combines depth information and skeletal data. A large amount of experiments are carried out both online and offline to test the effectiveness and real-time performance of the system. The experimental results indicate that our algorithm can quickly and accurately identify hand digit gestures of 0-9 and is comparable to advanced gesture recognition algorithms. The algorithm is not only invariant to complex background and illumination changes, but also satisfactory to reversal invariance, structural distortion invariance and rotation invariance.

With the development of human-computer interaction technology, gesture as a natural way for interaction is bound to become the future development trend. In this paper, some constraints are set in hand segmentation, which limits the natural characteristics of human-computer interaction. The future work should try to eliminate these constraints and make the hand segmentation algorithm more intelligent. Gesture recognition is still a difficulty in computer vision, meanwhile is also an important part of human-computer interaction. We plan to use machine learning methods for gesture recognition in our future work, expanding the gesture types for identification and improving the recognition accuracy .

6. Acknowledgements

This work is supported by National Natural Science Foundation of China (61463032, 51575412) and Scientific Research Foundation for Returned Scholars, Ministry of Education of China.

7. Reference

1. A. Erol, G. Bebis and M. Nicolescu, Vision-based hand pose estimation: A review, *Computer Vision and Image Understanding* **108**(1) (2007) 52-73.
2. K. M. Sagayam and D. J. Hemanth, *Virtual Reality*: *Hand posture and gesture recognition techniques for virtual reality applications: a survey*. (Springer-Verlag, 2016), pp. 1-17.
3. S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke and G. Rigoll, Static and dynamic hand-gesture recognition for augmented reality applications, in *ACM Int. Conf. Intelligent Multimodal Interaction Environments* (ACM Press, BeiJing, China, 2007), pp. 728-737
4. L. Riouxmaldague and P. Giguere, Sign Language Fingerspelling Classification from Depth and Color Images Using a Deep Belief Network, in *Canadian Conf. Computer and Robot Vision (CRV)* (IEEE Press, Montreal, Canada, 2014), pp. 92-97.
5. G. Diraco, A. Leone, and P. Siciliano, n active vision system for fall detection and posture recognition in elderly healthcare, in *Europe Conf. Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE Press, Dresden, Germany, 2010), pp. 1536-1541.
6. W. Gao, G. Fang and D. Zhao, A Chinese sign language recognition system based on SOFM/SRN/HMM, *Pattern Recognition* **37**(12) (2004) 2389-2402.
7. B. Y. L. Li, A. S. Mian, W. Liu and A. Krishna, Face recognition based on Kinect, *Pattern Analysis and Applications* **19**(4) (2015) 977-987.
8. G. Goswami, S. Bharadwaj, and M. Vatsa, On RGB-D face recognition using Kinect, *in IEEE Int. Conf. Biometrics: Theory, Applications and Systems (BTAS)* (IEEE Press, Arlington, USA, 2013), pp. 1-6.
9. M. Morana, *Advances onto the Internet of Things*: *3D Scene Reconstruction Using Kinect*. (Springer-Verlag, 2014), pp. 179-190.
10. Y. Li, Hand gesture recognition using Kinect, in *IEEE Int. Conf. Software Engineering and Service Science (ICSESS)* (IEEE Press, Beijing, China, 2012), pp. 196-199.
11. L. Bretzner, I. Laptev, and T. Lindeberg, Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering, in *IEEE Int. Conf. Automatic Face and Gesture*

Recognition (AFGR) (IEEE Press, Washington, USA, 2002), pp. 423-428.

12. C. Keskin, F. Kirac, Y.E. Kara, and L. Akarun, Real time hand pose estimation using depth sensors, in *IEEE Int. Conf. Computer Vision Workshops (ICCV Workshops)* (IEEE Press, Barcelona, Spain, 2011), pp. 1228-1234.

13. Z. Mo and U. Neumann, Real-time Hand Pose Recognition Using Low-Resolution Depth Images, in *IEEE Int.Conf. Computer Vision and Pattern Recognition (CVPR)* (IEEE Press, New York, USA, 2006), pp. 1499-1505.

vector, and the angles between these vectors are different. Assuming that the coordinates of the gravity center (x, y), the coordinates of two fingertips (x1, y2) and (x2, y2), and the angle of the two vectors θ, the angle's cosine of the two vectors can be computed by Eq. (1). After setting the corresponding angle threshold of these gestures, the actual value is compared with the threshold for accurate gesture recognition.nd choose the fingertip as the main feature. The gesture recognition process consists of the following steps: 1) Kinect V2 is used to acquire depth information and skeleton data; 2) the mask is obtained by hand segmentation based on the fusion of depth information and skeleton data; 3) the mask contour is extracted and the shape descriptor (fingertip) of the contour area is calculated; and 4) the features including the fingertips, angle between vectors and SIFT keypoints are combined to accurately and efficiently classify the gestures. The general framework of the system is given in Fig.5, and each step is described in detail in the following subsections.

c.