2014 IEEE 17th International Conference on Computational Science and Engineering

A Real-time Hand Gesture Recognition Approach Based on Motion Features of Feature Points

[1]Yingying She, [2]Qian Wang, [3]Yunzhe Jia, [4]Ting Gu, [5]Qun He, [6]*Baorong Yang
Software School
Xiamen University
Xiamen, China
[1]yingyingshe@xmu.edu.cn, [2]740409563@qq.com, [3]345019290@qq.com, [4]soliwly@qq.com, [5]378343237@qq.com
[6]ybr@xmu.edu.cn

***Abstract*—Dynamic hand gesture recognition enables people to communicate with computers naturally without any mechanical devices. Due to the spread of depth sensor such as Microsoft Kinect and Leap Motion, dynamic hand gesture recognition becomes possible for recognizing meticulous gesture information in real time. However, most of these methods recognize the hand gesture by fuzzy features such as contour size, which cause imprecise hand gesture recognition. This paper presents a precise tracing of feature points including palm center, fingertips and joints by using Kinect. A novel recognition method based on precise motion features of these feature points is also presented. Having been tested with a series of applications, our method is proved to be robust and effective, and suitable for further application in real-time HCI systems.**

Keywords—HCI, hand gesture recognition, motion feature, feature points

I. INTRODUCTION

Hand gestures recognition is one of the main areas of research for engineers, scientists and bioinformatics. With the popularity of motion capture devices such as the Microsoft Kinect and Leap Motion, the cost of depth information capturing has been greatly reduced. KinectSDK are used for hand position detection by detecting skeleton of human body in most Kinect-based applications. However, it treats the whole hand as one single point. One limitation of this approach is that people should be fully recognized. Furthermore, sophisticated hand gesture detection and recognition needs precise features like palm, fingers and fingertips, which inspires us to explore a method to capture detailed information of the hand using Kinect. In this paper, we present an approach of hand gesture recognition based on precise motion features of feature points like palm center and fingertips. This approach combines skeletal-based and appearance-based gesture recognition methodologies, processes hand gestures based on color and depth images, tracks hands movement and classify hand gestures in real-time.

II. RELATED WORK

HGR is the natural way of Human Machine interactions, and it can be applied to games, vision enabled robots, from virtual reality to smart home systems. There are many approaches associated with the accuracy of hand gesture recognition. Since the insufficient motion capture data is delivered by the KinectSDK, two major issues are the keys to success in hand gesture recognition: One is the hand region segmentation, and another is hand feature points positioning. Zhou Ren [1] focused on building a robust part-based hand gesture recognition system using the Kinect sensor. In that system, in order to get the hand shapes, Finger-Earth Mover's Distance (FEMD) was used to measure the dissimilarity between hand shapes. HiaathamHasan[2] explored a multi-layer neural network-based approach to recognize the hand gestures. Javier Molina[3] used static and dynamic models to get a real-time users independent hand gesture. Radhikacentre[4] presented a computer vision method to recognize the hand gesture from the image captured by a webcam.

In the processing of hand region segmentation, Dan Xu[5] proposed a method which can locate hands simultaneously in real-time by using skin-color detection and K-means in conjunction with stereoscopic depth information, and using a YCbCr color space filter to locate hand regions. The histogram color-based image threshold can be used to detect skin on the human body, and use the GMM model to segment human hand regions[6]. The morphological filtering technique can be used to effectively remove background and object noise from binary images[7]. Zhong Yang et al introduced the HSV color space skin filter[8]. Antonis A. Argyro et al [9] used a Bayesian classifier which is boot strapped with a small set of training data to detect skin-colored objects [10, 11].

In terms of positioning feature points, traditional approaches mainly focus on looking for the center of a maximum inscribed circle [12]. Jagdish L Raheja et al located the palm center by using distance transformation. However, the result of this method is not stable as we illustrate in the experimental result (section 4.2). After analyzing images captured by Kinect, we extend the idea presented [11] by positioning feature points based on morphological operations.

The Graham Scan is used in[10] to find the convex hull of a hand including the fingertips, but this process is not always precise due to the capture device factors and dynamic hand movements. Wang C[14] presented a hand gesture recognition algorithm by using the depth and skeleton from Kinect. Raheja, J.L.et al assumes that the fingertip is the point with minimum

*Corresponding author: Baorong Yang.

depth in each finger point cluster [13]. The problem with this method is it cannot precisely conduct certain fingertips when they are blending.

JongShill Lee et al. use the average distance between the point on the contour and the center point as the feature property of gesture recognition. Since the feature is relatively simple, just single float type, so they judge the gesture according to the value without using an additional classifier [18]. Manavender R. Malgireddy et al. propose a dynamic recognition framework using Sub-gesture model[19]. They describe the method of positioning gesture fragment by Sub-gesture and use the classifier based on HMM to experiment for zoom and rotate gesture. Zhong Yang uses a six-dimensional feature vector including Hand Position. Hand Velocity, Hand Size, Hand Shape and other feature information in the lecture[20]. He proposes a gesture fragment positioning method based on state and the data alignment method when using the classifier based on HMM.

III. Methodology

Feature points play an important role in the hand gesture recognition process. As in *Fig.1*, we develop a hand real-time hand gesture recognition system. In this paper, we explained methodologies for modules "Hand region detection along with feature points positioning" and "Feature points tracking and hand gesture recognition".

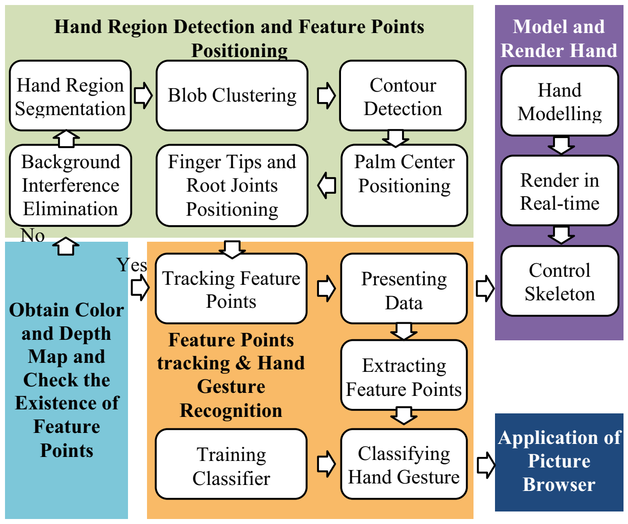

Hand Region Detection and Feature Points Positioning
Hand Region Segmentation
Blob Clustering
Contour Detection
Background Interference Elimination
Finger Tips and Root Joints Positioning
Palm Center Positioning
No
Yes
Obtain Color and Depth Map and Check the Existence of Feature Points
Tracking Feature Points
Presenting Data
Feature Points tracking & Hand Gesture Recognition
Extracting Feature Points
Training Classifier
Classifying Hand Gesture
Model and Render Hand
Hand Modelling
Render in Real-time
Control Skeleton
Application of Picture Browser

Fig. 1. The hand gesture recognition system

A. *Background Separation, Hand Region Extraction , Blob Clustering and Contour Detection*

The purpose of background separation is to extract the foreground target region and eliminate background interference from the original images. The background interference refers to noise points that are not in the hand region. In order to segment hand regions, we assume that hands are the closet objects to the Kinect. Based on the OTSU threshold method, we extract the targeted hand regions from depth images[15]. However, since the depth value of hands and nearby obstacles are similar, the OTSU method classifies these noises as the foreground targeted hand region, we use skin-color filter to separate hand regions and other obstacle noises. In addition, we fill in holes in hand regions based on the connected domain partition in the hands region.

After hand region extraction, we get the region for all hands. We have to separate each hand region in order to have accurate hand gesture recognition. We propose a clustering algorithm based on DBSCAN [16] (Density-Based Spatial Clustering of Applications with Noise) to do clustering.

After blob clustering, for every independent hand region, we have to do contour detection. Contour is an important element to express characteristics of gestures. The contour includes the external contour of the whole hand region and the internal contour in hand regions. In order to obtain the complete hand region contour while a finger and palm are overlapping, such as bending fingers, the calculation should count in the depth value.

B. *Palm Center Positioning*

The palm is one of the important indicators of hand movements. For each hand, we have to find its palm center as one of the feature points. In our project, the palm center positioning algorithm is based on morphology. The morphology operation approach is extended from [7]. The idea is to use the method of corrosion morphology to process hand region, remove fingers form the hand region, and calculate the palm centre based on the leaving pixels in the hand. In general, more distance from the hand to the Kinect, less pixels in the hand and its fingers. So, the region of the corrosion operator in the hand area is related to the total number of pixels in hand. Our morphological operation based palm center locating algorithm uses the rectangular corrosion operator to calculate the palm center.

C. *Fingertips and Finger root Positioning*

The way to locate the fingertip is by looking for the point with the largest curvature in the finger. We use the geometric feature of points in the finger to get the fingertip candidates. First, we set a threshold, a point and vectors in the contour. If the angle between these vectors is less than the point, it is the candidate of a fingertip. Usually, the candidate point set is an arc in the contour.

$$\text{Cdd} = \left\{ p_t \middle| p_t \in Cont, < \overrightarrow{p_t p_{t\text{-}m}} 、 \overrightarrow{p_t p_{t+m}} > < \theta \right\} \tag{1}$$

After getting the candidate set, we have to do clustering, and obtain independent arc segments. Then, if the independent fingertip arc is the ***jth*** point in the arc, then candidate fingertips are denoted in yellow arcs as in *Fig. 2*.

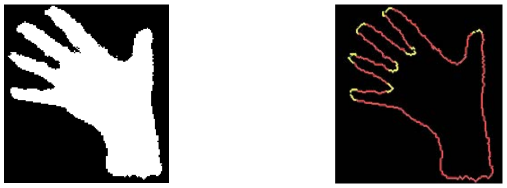


Fig. 2. Candidate set

In general, arc C_i (i=1,2, ⋯, n) is a symmetrical arc. The fingertip point is the mid-point in the arc. However, in practice, since the resolution of Kinect and different interaction operations, arcs are not always symmetrical. In order to get the accurate fingertip coordinate values, we calculate the angle between mid-point and its neighbor points which has a distance *m* within the arc. In total, there are *2m+1* neighbor points corresponding to *2m+1* angles. We choose the point with the smallest angle as the fingertip candidate.

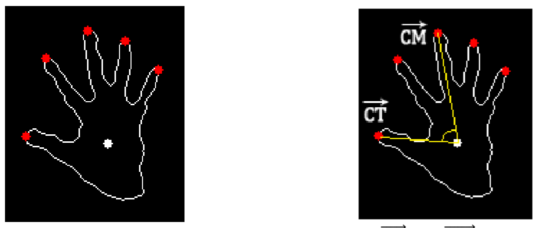
$$D_i=\left\{p_{i,j}\middle|p_{i,j}\in C_i,\frac{C_i.size}{2}-m<j<\frac{C_i.size}{2}+m\right\} \tag{2}$$

$$FingerCdd_i=argmin_{p_t\in D_i}<\overrightarrow{p_tp_{t-m}},\overrightarrow{p_tp_{t+m}}> \tag{3}$$

Here is the algorithm to find fingertip candidates *fingerCdd.*

```
Algorithm: Finding Fingertip Candidate
Input: hand contour contour, threshold $\theta$, $\tau$
Output: fingertip candidates fingerCdd
1. sort all points in contour;
2. for $i^{th}$ point $p_i$ in contour
       calculate angle $\gamma$ between $\overrightarrow{p_ip_{i-m}}$, $\overrightarrow{p_ip_{i+m}}$; ($m$=10 in our experiment)
       if $\gamma < \theta$
           add $p_i$ to the end of candidate
   end for
3. cluster candidate into $C_1 \cup C_2 \cup \ldots \cup C_n$ by distance between points in candidate;
4. add the point with lowest corresponding $\gamma$ in each cluster $C_1, C_2, \ldots, C_n$ to fingerCdd;
5. return fingerCdd;
```

There are fingertip and finger webs in fingertip candidates. We have to filter the finger web point, and conduct the fingertip points. When opening five fingers, the distance between the fingertips and the palm center is longer than the distance from the finger web to the palm center. Base on this idea, we filter all fingertips. The process of labeling fingertips is conducted under the assumption that all five fingertips have been detected, which means *fingertipssize*=5. In this process, the fingertips will be marked: thumb fingertip (*T*), index fingertip (*I*), middle fingertip (*M*), ring fingertip (*R*) and little fingertip (*L*).When five fingers are open, starting from palm centre (*C*) the five vectors hold where it denotes the intersection angle of the angle between fingertips, as shown in *Fig.3*.



(a) Fingertips (b) $\overrightarrow{CT}$ and $\overrightarrow{CM}$

Fig. 3. The angle of $\overrightarrow{CT}$ and $\overrightarrow{CM}$

The distance between palm center and thumb fingertip and the distance between palm center and little fingertip are smaller than the distances between palm center and the other three fingertips. Therefore, we can get *T* and *L* first. Then distinguish them by the assumption that the thumb fingertip is thicker than the little fingertip.

The algorithm of labeling fingertips is described as below:

```
Algorithm: Labelling Fingertips
Input: fingertips points fingertips
Output: labelled fingertips (from thumb to little finger) fingertips
1. calculate distances between center and each point in fingertips, select $f_1$ and $f_2$ with lowest and second lowest distance;
2. calculate $\alpha$、$\beta$, former is angle between $f_1$ and m-th ahead and after points in contour, using same method in 3.5.1, $\beta$ is for $f_2$ as well;
3. if $\alpha>\beta$, label $f_1$ as $T$ (thumb), otherwise label $f_2$ as $T$ (thumb);
4. calculate $<\overrightarrow{CenterT},\overrightarrow{CenterFingertips_i}>$, where $Fingertips_i$ are other four fingertips, sorting them in ascend order, then label the fingertips as $I$ (index finger), $M$ (middle finger), $R$ (ring finger), $L$(little finger) in that order;
5. return labelled fingertips;
```

Finger root joint is defined as the intersection of fingers middle line and the palm (as indicated as yellow points in *Fig. 4*). Moreover, the distance between joint and palm centeris roughly equal to the distance between finger valley and palm center. Define is when one of the fingers middle line, d is the mean of distances from palm center to five finger valleys. Then the corresponding root joint (*x*, *y*) holds.

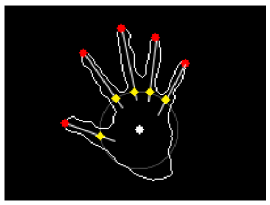


Fig. 4. Finger root joints

$$\begin{cases} Ax+B=Y \\ (x\text{-}center.x)^2+(y\text{-}center.y)^2=d^2 \end{cases} \tag{4}$$

Suppose there is a fingertip on the sorted contour，then we choose another n (n is a threshold) pair of points.Then calculate the middle points of all the pairs, the middle line of the finger is calculated by using the Min Square algorithm, as in (5).

$$\begin{cases} \bar{t}=\frac{1}{n}\sum_{j=1}^{n}t_j \\ \bar{s}=\frac{1}{n}\sum_{j=1}^{n}s_j \\ A_i=\frac{\sum_{j=1}^{n}t_js_j-n*\bar{t}*\bar{s}}{\sum_{j=1}^{n}t_j^2-n\cdot(\bar{t})^2} \\ B_i=\bar{s}-A_i\bar{t} \end{cases} \quad i=1,2,3,4,5 \tag{5}$$

The whole algorithm is described as below:

```
Algorithm: Locating Finger joints
Input: fingertips points fingertips, palm centrecenter, sorted contour contour, finger valleys fingervalleys
Output: finger joints joints
1. calculate distances between center and each point in fingervalleys, then calculate mean of these distances;
2. for each $f_i$ in fingertips
    get points $mid_1, mid_2, \ldots, mid_n$ ($n$=30 in out experiment), where $mid_j$ is middle point of jth ahead and after points of $f_i$ in contour;
    calculate $line_i$ using Min Square with middle points;
    calculate intersection of $line_i$ and $(x\text{-}center.x)^2 + (y\text{-}center.y)^2 = mean^2$, then add it to joints;
  end for
3. return joints;
```

D. Feature Points Tracking

When hand feature points of a certain frame were detected, we begin to tracking them in the next frame. That is, the most likely corresponding feature points in next frame are predicted through the positions of feature points in current frame without re-detection.

1) Displacement-based tracing of palm center and finger roots

The position change of feature points is basically demonstrated as panning of points during the gesture-based interactions. The position of palm center is calculated from the corroded hand area by distance transform in the same detection algorithm. It is assumed that the relative positions of the finger roots and palm center are fixed, and the movements of fingers do not affect positions of finger roots. We take the position of palm center from previous frame as $prePalmCent$, and positions of finger roots as $preMCP_i$ (i=1,2,3,4,5). After the palm center position of current frame $curPalmCent$ is detected, the offset of palm center is calculated as in (6).

$$\begin{cases} offset_x = curPalmCent.x\text{-}prePalmCent.x \\ offset_y = curPalmCent.y\text{-}prePalmCent.y \end{cases} \tag{6}$$

And the new finger roots position of current frame are calculated as in (7).

$$\begin{cases} curMCP_i.x = preMCP_i.x + offset_x \\ curMCP_i.y = preMCP_i.y + offset_y \end{cases} \quad (i=1,2,3,4,5) \tag{7}$$

2) Prediction-based tracing of fingertips

Fingertips are the most important parts in gesture-based interaction, and the most complicate parts in tracking. Accuracy and continuity are the uppermost standards while tracking fingertips. A break off of fingertips tracking in a certain frame will fail the tracking in the next frame, and eternally stop the tracking process. Thus, fingertips should be located precisely and continuously in every single frame. The feature point position of previous frame is used for predicting the likely position of corresponding feature point in current frame, and confirming the exact position of feature point by current frame. The process is explained in the following steps:

- Predict the possible positions of fingertips in current frame, which are the colored areas in *Fig.5*(a) , ranging 10 pixels from each fingertip predicted in previous frame.
- Find a candidate set of feature points C_i corresponding to the i^{th} feature point in the previous frame (the coloured area in *Fig.5* (c))among the points of hand region shown in *Fig.5*(b) in current frame, which is the intersection of the valid pixels of current frame and the predicted set of fingertips positions from the previous frame.
- Take the point with most highest possibility C_i as the i^{th} feature point (the finger tips) of current frame. The new fingertips are marked in *Fig.5*(d).

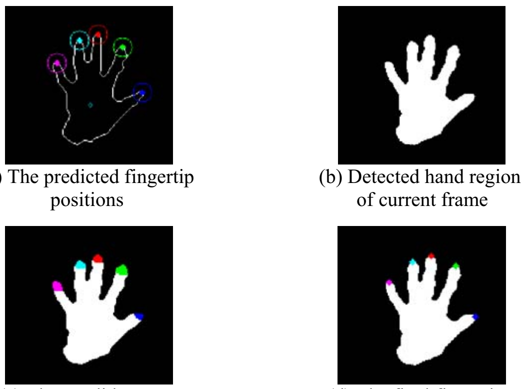

(a) The predicted fingertip positions
(b) Detected hand region of current frame
(c) The candidate sets
(d) The final fingertips

Fig. 5. Prediction-based tracking process of fingertips

E. Feature Extraction

1) Feature extraction based on velocity of feature points

In dynamic hand gesture recognition, feature vectors reveal the motion features of series of gesture sequences. In order to extract the velocity feature from feature points, we take the offset of distances between fingertips and palm center in two frames as the velocity of fingertips. A negative direction means decrease of relative distance, while a positive direction means increase of relative distance.

Palm velocity is defined as the displacement of palm center in two frames. Positive value means moving towards left and negative value means moving towards right. If taking the position of fingertip in previous frame *as* $preFintip_i$ (i=1, 2, 3, 4, 5), the palm center of previous frame as $prePalmCent$, the position of fingertip in current frame as $curFintip_i$ (i=1, 2, 3, 4 ,5) and the palm center of current frame as $curPalmCent$, the velocity of fingertip VF_i and velocity of palm center VP can be computed as in (8) to (11).

$$preDis_i = EuclidDis(preFintip_i, prePalmCent) \tag{8}$$

$$curDis_i = EuclidDis(curFintip_i, curPalmCent) \tag{9}$$

$$VF_i = curDis_i\text{-}preDis_i \tag{10}$$

$$VP = \begin{cases} EuclidDis(curPalmCent, prePalmCent) \\ \quad \text{if } prePalmCent.x \geq curPalmCent.x \\ \\ -EuclidDis(curPalmCent, prePalmCent) \\ \quad \text{if } prePalmCent.x < curPalmCent.x \end{cases} \quad (11)$$

The *EuclidDis(a,b)* calculates the euclidean distance between *a* and *b*.

For adjacent two frames, we define a feature combination vector as

$$\vec{F_t} = <VF_1, VF_2, VF_3, VF_4, VF_5, VP>, \quad (12)$$

in which *VP* is the velocity of palm center, and other five vectors are velocity of the five finger tips.

2) Feature extraction based on fuzzy hand region

The feature vector extraction of hand region is based on fuzzy feature of hand. The geometric description of the whole hand region is used to do the extraction. The fuzzy feature vector

$$\vec{F} = <A_t, V_t, \vec{S_t}> \quad (13)$$

between adjacent frames $frame_t$ and $frame_{t+1}$ is extracted based on the following three features, the area change feature of the rectangular region of hand A_t, velocity of the center of hand region V_t, and the shape description operator S_t[10].

F. Classification and Identification

In dynamic hand gestures recognition the primary problem is to find the start and finish points in the image sequence. It is the abstraction of the hand gesture segment from continuous image sequence. We use the algorithm introduced in [20] to locate the gesture sequence in continuous image sequences.

After we obtained the hand gesture image sequences, the gesture classification process identify different gestures. In general, there are two categories of gesture classification algorithms, one is overall classification and another is partial classification. The hidden markov classification algorithm belongs to the overall classification algorithm. It abstracts pattern vector sets $(\vec{F_1}, \vec{F_2}, \ldots, \vec{F_n})$ from the gesture sequence and uses them as input of the algorithm. The idea is to use an entire vector set as the observed sequence, and find the corresponding Hidden Markov Model. Algorithms based on the naive Bayesian algorithm and forward feedback neural network algorithms are partial classifications. They classify each pattern vector $\vec{F}$, obtain the classification sets $(Gesture_{\vec{F_1}}, Gesture_{\vec{F_2}}, \ldots, Gesture_{\vec{F_n}})$, and find the corresponding gesture based on training sets. For these three classification algorithms, the gesture classification accuracy depends on the scale and effectiveness of the training data sets.

IV. Experiments

A. Experiment of Locating and Labelling Fingertips and Joints

Analyzing feature points of hand includes locating fingertips and joints, and labelling them. In this test, we design two tests. The first testis for locating, and the second one is for labeling.

In the first test, we detect fingertips in five different gestures; in the second test, labelling is done based on the results of gesture with five fingers open. The parameters m=20 and $\theta = 55°$. *Fig.6* denotes the process and results of locating fingertips. The first column is the captured images; the second column is the contour and palm center; the third column shows the fingertips candidate; the last column is the results of fingertips.

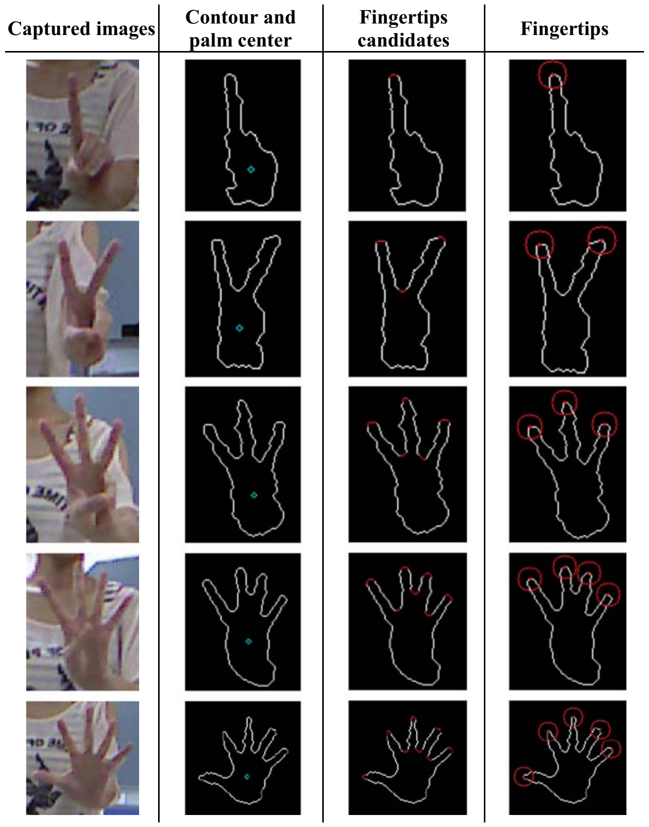
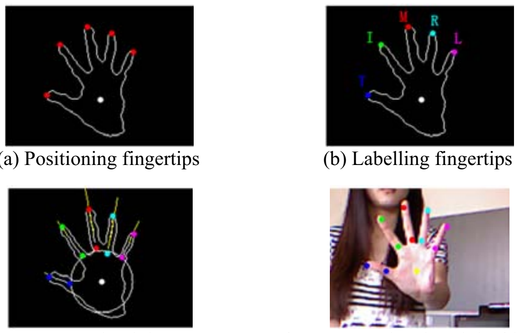


Fig. 6. Result of locating fingertip

Fig.7 shows the detected feature points. Based on the result of detecting fingertips, the proposed algorithm finds thumb (blue point in *Fig.7*) first. The other four fingertips are labelled based on the relative position with thumb. Finger root joints are found with Min Square algorithm.



(a) Positioning fingertips (b) Labelling fingertips

(c) Detecting joints (d) Feature points in original image

Fig. 7. Result of labelling fingertips and finger root joints

B. Finger Root Joint Tracking Experiments Based on Related Positions

In this experiment, we assume three testing scenarios after feature points detected in the initial frame(*Fig.8*(a)).

1) Hand panning(*Fig.*8(b))*;*
2) Hand keeps static and fingertips keeps moving(Fig.8(c));
3) Palm rollover(*Fig.8*(d)).

The testing results are denoted in *Fig.8*. In the *1st* and *2nd* scenarios, the hand root joints could be positioned accurately. The reason is the relative positions of palms and finger roots are relative stable. However, in the *3rd* scenario, the finger root tracking effect is not as our expectation. The result has certain excursion from the right position. The reason for the recursion is the relative position between palm center and finger root is changed when the tester rollovers the hand.

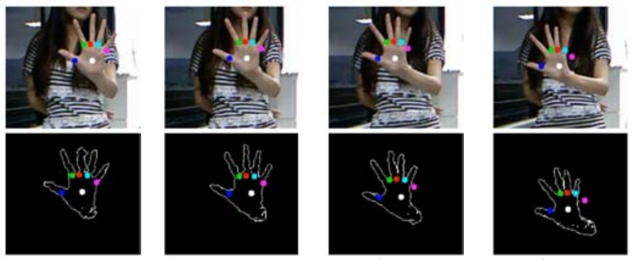

(a) initial positions (b) *1st* scenario: panning (c) *2nd* scenario: moving fingertips (d) *3rd* scenario: palm rollover

Fig. 8. Result of tracing finger roots in four scenario

C. Fingertip Tracking Experiments Based on the Prediction Model

The fingertips tracking are the most difficult part in the whole process of interaction. It is also the key to the success of the hand gesture recognition. In our experiment, we test in four fingertips tracking scenarios, such as fingers curling, scattered, translation, rotation denoted in *Fig.9.* At the same time, we test fingertips tracking when finger information lost in certain frames caused by hardware. The fingertip tracking model is based on the depth information combined with 2D distance. The depth compensation value is 5*mm*, and the 2D screening radius is 10*mm*. In *Fig.9*, we use different colors to denote different fingertips. Blue is for the thumb, green is for the index finger, red is for the middle finger, green is for the ring finger and magenta is for the little finger. We could get satisfied testing results from Form these four scenarios. And, the finger information lost would not affect the later fingertip tracking, such as the *2nd* and *4th* frame in the translation scenario.

D. Gestures classification

We defined four gestures in this test, including zoom in, zoom out, left translation and right translation. Three testers repeat four gestures thirty times, and testing the accuracy of hand gesture recognition in the following three feature combinations.

$$\overrightarrow{F_1} = <VF_1, VF_2, VF_3, VF_4, VF_5> \tag{14}$$

$$\overrightarrow{F_2} = <VF_1, VF_2, VF_3, VF_4, VF_5, VP> \tag{15}$$

$$\overrightarrow{F_3} = <VF_1, VF_2, VF_3, VF_4, VF_5, VP, VP, VP> \tag{16}$$

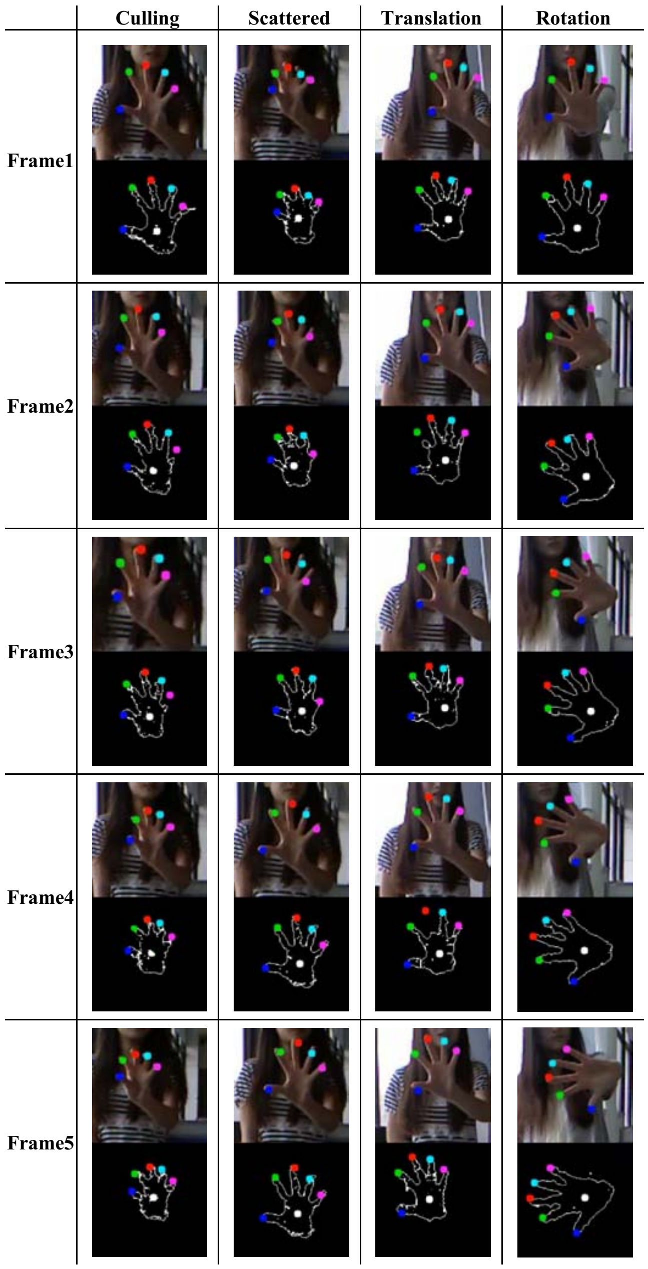


Fig. 9. Result of tracing fingertips

We apply the naïve Bayesian algorithm and forward feedback neural network algorithm (take FFNN for short) on each of the three feature combination vectors, and the result shows that the second feature combination has the highest accuracy for hand gesture classification, as shown in *Fig.10*. Then we apply the two algorithms on the second feature combination vector. And the result in *Table I* shows that the naïve Bayesian classification algorithm has better performance than the forward feedback neural network algorithm.

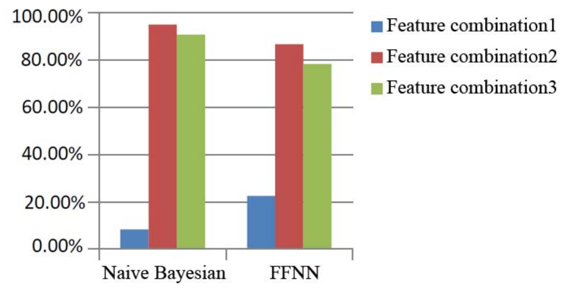


Fig. 10. Gesture classification algorithms comparison results

TABLE I. RECOGNITION RESULT BY THE TWO METHODS

Hand Gesture	**Number of Test**	**Naïve Bayesian**		**FFNN**	
		Number of Successful Recognition	***Recognition Rate***	***Number of Successful Recognition***	***Recognition Rate***
Zoom In	30	28	93.33%	29	96.67%
Zoom Out	30	28	93.33%	16	53.33%
Left Translation	30	29	96.67%	30	100.00%
Right Translation	30	29	96.67%	29	96.67%
Total	120	114	95.00%	85	86.67%

V. Conclusion

This paper focuses on the work of hand gesture recognition based on motion features of feature points in hands. Experiments of this approach are conducted to prove the stability and robust. Extended works of this approach is tracking feature points for multi-hands, generating the hand gesture library and embedding into the hand gesture recognition system. This approach could be extended to HCI and VR (Virtual Reality) / AR (Augmented Reality) systems such as a sign language recognition system, immersive games and other NUI (Nature-User-Interaction) systems.

Acknowledgment

The authors would like to acknowledge the support of Education game platform based on mobile cloud services (3502Z2014116, principal, technological innovation projects for early-stage companies, Science and Technology Planning of Xiamen, 2014.6-2016.6).

References

[1] Zh. Ren, J.S. Yuan, J.J. Meng, Zh.Y. Zhang, “Robust part-based hand gesture recognition using kinect sensor”. IEEE Transactions on Multimedia,vol. 15 (5), pp. 1110-1120, Feb. 2013

[2] H. Hasan. and S. Abdul-Kareem, “Static hand gesture recognition using neural networks,” Artificial Intelligence Review, vol. 41(2), pp. 147-181, Feb. 2014

[3] M. Javier et al., “Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models,” Machine vision and applications, vol. 24(1), pp. 187-204, Jan. 2013

[4] Bhatt, R., N. Fernandes, and A. Dhage, “Vision based hand gesture recognition for human computer interaction: a survey,” International Journal of Engineering Science and Innovative Technology, vol. 2(3), Nov. 2012

[5] D. Xu, Y.L. Chen, X.Y. Wu, Y.SH Ou and Y.SH Xu, “Integrated approach of skin-color detection and depth information for hand and face localization,” Proceedings of 2011 IEEE International Conference on Robotics and Biomimetics, Karon Beach, Phuket, pp. 952-956, Dec. 2011

[6] Ming, Y., Q. Ruan, and A.G. Hauptmann, “Activity recognition from rgb-d camera with 3d local spatio-temporal features,” Proceedings of 2012 IEEE International Conference on Multimedia and Expo, Melbourne, VIC, pp.344-349, July 2012

[7] Ghosh, D.K. and S. Ari, “A static hand gesture recognition algorithm using k-mean based radial basis function neural network,” Proceedings of 2011 8th International Conference on Information, Communications and Signal Processing, Singapore , pp. 1-5, Dec. 2011

[8] Zh. Y., Y. Li, W.D. Chen, Y. Zheng, “Dynamic hand gesture recognition using hidden Markov models,” Proceedings of 2012 7th International Conference on Computer Science & Education, Melbourne, VIC, pp. 360-365, July 2012

[9] A.A. Argyros, and M.I. Lourakis, “Real-time tracking of multiple skin-colored objects with a possibly moving camera,” Proceedings of 8th European Conference on Computer Vision, Prague, Czech Republic, pp. 368-379, May 2004

[10] Y. Wen, Ch. Y. Hu, G.H. Yu, Ch. B. Wang. “A robust method of detecting hand gestures using depth sensors,” 2012 IEEE International Workshop on Haptic Audio Visual Environments and Games, Munich , pp. 72-77, Oct. 2012

[11] M.K. Bhuyan, D.R. Neog, and M.K. Kar, “Hand pose recognition using geometric features,” Proceedings of 2011 National Conference on Communications, Bangalore, pp. 1-5, Jan. 2011

[12] H. Liang, J.S. Yuan, and D. Thalmann, “3D fingertip and palm tracking in depth image sequences,” Proceedings of the 20th ACM international conference on Multimedia, Nara, Japan, pp. 785-788, Oct. 2012

[13] C. Wang, and S. Chan, “A new hand gesture recognition algorithm based on joint color-depth Superpixel Earth Mover's Distance,” 2014 4th International Workshop on Cognitive Information Processing, Copenhagen, Copenhagen, pp. 1-6, May 2014

[14] J.L. Raheja, A. Chaudhary, and K. Singal, “Tracking of fingertips and centers of palm using kinect,” Proceedings of 2011 Third International Conference on Computational Intelligence, Modelling and Simulation, Langkawi, pp. 248-252, Sept. 2011

[15] A. Kurakin, Z. Zhang, and Z. Liu, “A real time system for dynamic hand gesture recognition with a depth sensor,” Proceedings of the 20th European Signal Processing Conference, Bucharest, pp. 1975-1979, Aug. 2012

[16] M. Ester, H.-P. Kriegel, J. S., X.W. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, pp. 226-231, Aug. 1996

[17] S. Suzuki, “Topological structural analysis of digitized binary images by border following,” Computer Vision, Graphics, and Image Processing, vol. 30(1), pp. 32-46, April 1985

[18] J.S. Lee, Y.J. Lee, E.H. Lee, and S.H. Hong, “Hand region extraction and Gesture recognition from video stream with complex background through entropy analysis,” Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, CA, USA. pp. 1513-1516, Sept. 2004

[19] M.R.Malgireddy, J.J. Corso, S. Setlur, V. Govindaraju and D. Mandalapu, “A Framework for Hand Gesture Recognition and Spotting Using Sub-gesture Modeling,” Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, pp. 3780-3783, Aug. 2010

[20] Zh. Yang, Y. Li, W.D. Chen, and Y. Zheng, “Dynamic Hand Gesture Recognition Using Hidden Markov Models, ” Proceedings of 2012 the 7th International Conference on Computer Science & Education, Melbourne, VIC. pp.360 - 365, July 2012