Wrist Recognition and the Center of the Palm Estimation Based on Depth Camera

Zhengwei Yao, Zhigeng Pan, Shuchang Xu
Digital Media and HCI Research Center
Hangzhou Normal University
Hangzhou, China, 310036
yzwiron@hznu.edu.cn

Abstract—When the hand and the forearm enter the available depth range of the depth camera, the data of the hand and the forearm will be extracted together. If process these data as a whole, which maybe affect some important algorithms such as the center of the palm estimation, the orientation of the hand estimation and hand tracing. The paper analyzes the motion features of the wrist and the contour features of the hand, takes advantage of the geometric characteristics of an inscribed rectangle and proposes a wrist recognition algorithm. In order to reduce the computing time of estimating the center of the palm, the paper analyzes the geometric characteristics of an acute triangle and an inscribed circle, combines the features of the hand interaction and proposes a new algorithm of estimating the center of the palm. Proved by the experiments, the wrist recognition algorithm can separate the hand from the forearm well, and the new algorithm of estimating the center of the palm has a distinct advantage over the original algorithms in the performance.

Keywords—Wrist recognition; the center of the palm estimation; depth camera; hand segment; gesture interaction

I. INTRODUCTION

In recent decades HCI is an important research area focused by many researchers. The goal of HCI is to realize the natural intuitive interaction. That means people don’t need to intentionally learn how to control a system, and they can directly communicate with a system based on the knowledge they learned in the daily life[1]. Hand is one of the cleverest control tools for human, so hand has been widely applied in HCI, especially in virtual reality and augmented reality fields[2][3]. Based on the input devices that are used for feeding the model, these gesture interaction systems can be classified into four classes: data glove, marker[4][5], hand image[6][7][8], and drawing[9][10]. The system presented in this paper belongs to the third class.

In this class the hand segment and extraction plays a major role for the successful application of pending processes to validate the hand recognition and is considered the most difficult ring of this processing chain[11]. The complexity of scene and light directly affect the hand segment for video-based gesture system[12]. Depth camera can solve some problems the video cameras suffer in the conditions like low light and other skin-colored objects in a scene. It is common to simply use a depth threshold to isolate the hands[13][14]. Depth threshold determines the hands to be those points between some near and far distance thresholds around the Z (depth) value—which has to be predetermined and instructed to the user. One modification to depth threshold is to determine the predicted hand depth according to the location of other body parts, rather than assuming the hand is necessarily the closest object in the scene. Cerlinca[15] and Van den Bergh[16] both use the robust face detector to determine the head location and then estimate candidate hand locations. Fujimura[17] uses body detection based on Karhunen-Loeve Decomposition to estimate hand locations.

Whatever method is used, it is difficult to find the wrist to isolate the hand from the forearm. The way proposed by Z. Ren et al.[14] requires a user to wear wristband in order to isolate the hand from the forearm. The method proposed by Lee et al.[18] can find the fingertips and the center of the palm without needing to separate the hand from the forearm. But when a large part of the forearm enters the valid depth range, this method may not successfully estimate the center of the palm. The center of the palm is an important data in a gesture interaction system: the line through the center of the hand cluster and the center of the palm can also be used as a hand orientation indicator, and the center of the palm is usually used to represent a mouse in an air mouse system or in an air multi-touch system. The paper proposes a wrist recognition algorithm applied in the hand segment. That is the basis for the pending estimating the center of the palm. The paper also proposes a new method of estimating the center of the palm. This new method takes advantage of the geometric features of an acute triangle and a maximum inscribed circle, which improves the performance of the system.

II. HAND SEGMENT AND THE CENTER OF THE PALM ESTIMATION IN THE GESTURE SYSTEM

The system presented in this paper is built on the Candescant NUI project. The Candescant NUI project uses the OpenNI framework to extract depth data from the Kinect sensor. Based on the depth threshold predetermined, it isolates the hand from the background and uses the clustering algorithm to classify the hand data. If two hands are used in the interaction, the clustering algorithm will generate two clusters; otherwise, there is only one. Then the convex hull containing the hand is calculated, and the contour of the hand is traced. By examining the convex hull and the contour, the fingertips and the center of the palm are detected. As shown in Fig. 1.
1) Cluster the hand data: a depth threshold is manually set in advance to specify the depth range (0.5m-0.8m). Objects in this range can be detected. A hand pixel may be referred to as a point. These hand pixels are projected to a 2D space. Then the K-means clustering algorithm is applied to partition these pixels into two clusters according to their positions in the 2D space.

2) Compute the Convex hull of the hands: the Graham scan algorithm is used to compute the convex hull of the detected hand clusters.

3) Detect the hand contours: the hand contours are detected by the modified Moore-Neighbor tracing algorithm.

4) Detect the fingertips: find all candidate points ($C$) that are both on the convex hull and the contour. For each point $p_0$ in $C$ take the two points $p_1$ and $p_2$ in the two opposite directions along the contour that are at a given distance to $p_0$. Find the midpoint of $p_1$ and $p_2$, and calculate the distance between the midpoint and $p_0$. If the distance is greater than a specified threshold, the $p_0$ is identified as a fingertip.

5) Estimate the center of the palm: this is done by finding the biggest circle inside the hand contour. Find all candidates points inside the hand contour. For each candidate point, calculate the smallest distance to any point in the contour, then take the candidate point with the largest minimum distance as the center of the palm. Actually for performance reasons, the system takes every 8th point to speed up.

The center of the palm is pretty important information for gesture interaction, because it is quite stable during rotating, opening and closing the hand. However when a large part of the forearm enters the available depth range, the original algorithm regards the forearm and the hand as a whole, which may result in the estimated center of the palm located in the forearm. In addition, although the original algorithm of the center of the palm estimation is easy to understand, its performance still needs to be improved. The paper proposes a wrist recognition algorithm and a new algorithm of estimating the center of the palm to solve these problems.

III. HAND SEGMENT

A. Wrist Recognition

It can be seen from the contour of hand that the wrist points should be pit points, and it is clear from the moving characters of the wrist that the wrist can rotate enormously. So the contour of the wrist can be changed from moment to moment during the interaction. At one moment, there is maybe only one wrist point detected in the hand contour, as shown in Fig. 2. The method proposed in this paper is to find an obvious pit point in the contour of hand and then create an appropriate inscribed rectangle to find another wrist point. The detailed steps are shown below:

Step 1: find all points that are both on the convex hull and the contour of the hand. According to the sequence order of these points, define lines (each line’s ends are two consecutive points). Based on the following laws, find the candidate lines from these lines:

1) The ends of the candidate line should not be both fingertips.

2) The distance of the candidate line should not be less than a specific value.

As shown in Fig. 3, $p_0p_1$, $p_5p_6$ and $p_6p_7$ are the candidate lines.

Step 2: find the corresponding candidate contours whose ends are the ends of the candidate lines. If there are other intersections of the convex hull and the contour in the candidate contour, that means the searching direction of the candidate contour is opposite and should be changed.

Step 3: calculate the maximum distance between the candidate line and the corresponding candidate contour. If there are several candidate lines, get the largest distance from these maximum distances. The point with this largest distance is one of the wrist points ($w_1$ in Fig. 3).
Step 4: connect this wrist point to each point in the hand contour, and take these connecting lines as the diagonals of rectangles. If the rectangle is not inside the hand contour, the corresponding point in the contour is not another wrist point; otherwise, take this point as a candidate point, and compute the distance of the rectangle diagonal. In all candidate points, find out the point with the shortest rectangle diagonal. That point is another wrist point \((w^2\text{ in Fig. 3})\).

Determine if a rectangle is inside the hand contour: for each point in the hand contour, determine if the point is inside the rectangle. If all of the points are not inside the rectangle, the rectangle is inside the hand contour, vice versa.

Determine if a point is inside the rectangle: classify the four sides of the rectangle into two groups. Each group consists of two parallel lines. If the point is located in the same side of the lines of one group, this point is outside the rectangle. Otherwise, determine the positional relationship between the point and another group’s lines. As long as the point is not located in the same side of the lines of the both groups, the point is inside the rectangle.

**B. Optimize the Hand Contour**

After finding the two wrist points, the hand contour can be divided into two parts by the wrist points. If the part is located in the same side of the wrist as the last center of the palm estimated, this part is the optimized hand contour.

**IV. ALGORITHM OF ESTIMATING THE CENTER OF THE PALM**

The algorithm proposed in this paper refers to the Sun’s[19] idea to improve the efficiency of estimating the center of the palm. However the Sun’s method is good only for the smooth contours. The shape of the hand is complex, and Kinect sensor is sensitive to noise. An error has been experimentally found that the center of the palm is easily located in a finger by mistake. As a result, the Sun’s method can not be directly applied in the system of this paper.

Actually the gesture is not a kind of high-precision interaction, so it is not necessary to find the biggest inscribed circle inside the hand contour. In other words, the center of the palm estimation algorithm can find a circle approaching the biggest inscribed circle – it is permitted that a very few points in the hand contour are located inside the circle found. Moreover, in order to avoid the center of the palm being located in a finger by mistake, the minimum radius of the palm is used as a threshold. The new algorithm of estimating the center of the palm is shown below:

**Step 1:** select three points from the hand contour \((P_1, P_2\) and \(P_3\) in Fig. 4). The three points form an acute triangle.

**Step 2:** obtain two intersectant chords from this acute triangle \((P_1P_3\) and \(P_2P_3\) in Fig. 4), find the intersection point \((O_j\) in Fig. 4) of the vertical bisectors of these two intersectant chords – this intersection point is a candidate center of the circle, and calculate the coordinate values, \(X\) and \(Y\), of the \(O_j\) using (1) and (2):

\[
X_j = \frac{(Y_y - Y_1)(Y_3 - Y_j)(X_1 - X_2) + X_u(X_3 - X_1)(Y_2 - Y_j) - X_u(X_3 - X_2)(Y_2 - Y_1)}{(X_3 - X_1)(Y_j - Y_2) - (X_3 - X_2)(Y_j - Y_1)} \\
Y_j = \frac{(X_x - X_1)(X_3 - X_j)(X_3 - X_2) - X_u(Y_3 - Y_1)(X_2 - X_j) + X_u(Y_3 - Y_2)(X_3 - X_1)}{(X_3 - X_1)(Y_j - Y_2) - (X_3 - X_2)(Y_j - Y_1)}
\]

(1) and (2)

Hereinto, \(X_j\) and \(Y_j\) are the coordinate values of \(O_j\). \(X_1\) and \(Y_1\) are the coordinate values of \(P_1\). \(X_2\) and \(Y_2\) are the coordinate values of \(P_2\). \(X_3\) and \(Y_3\) are the coordinate values of \(P_3\). \(X_u\) and \(Y_u\) are the coordinate values of the midpoint of the chord \(P_1P_3\). \(X_N\) and \(Y_N\) are the coordinate values of the midpoint of the chord \(P_2P_3\).

Use the (3) to calculate the radius \(R_j\) of the circle that is in the distance from the center to these three points respectively.

\[
R_j = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \quad (3)
\]

Step 3: use the (4) to calculate the distances from each point in the hand contour to the center \(O_j\).

\[
R_j = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \quad (4)
\]

If the number of the distances that are less than the radius \(R_j\) of the circle is less than the value of the threshold inside (how many points in the contour are located inside the circle), and the \(R_j\) is greater than the value of the threshold \(minR\) (the minimum radius of the palm), the center of the circle is the center of the palm. Otherwise enter step 4.

Hereinto, \(i = 1, 2, \ldots, n\); \(X_i\) and \(Y_i\) are the coordinate values of the point in the hand contour; the variable \(n\) represents the number of points in the hand contour.

Step 4: one end-point of these two intersectant chords is replaced by point \(P_{min}\) and the three end-points of the intersectant chords must form an acute triangle at all time. For example point \(P_3\) is replaced by point \(P_{min}\) in Fig. 4. The overlap segment from step 2 to step 4 is run repeatedly until the ending condition is true (find a circle approaching the biggest inscribed circle).

**V. RESULTS AND DISCUSSION**

The system presented in this paper uses the OpenNI framework to extract depth data from the Kinect sensor. The
proposed approaches have been tested on the PC with AMD Athlon(tm)64*2 Dual Core Processor 5200+ 2.7GHz CPU, 4GB RAM, NVIDIA GeForce 9600GT Graphics card and Window7 32bit OS. The number of the hand clusters in the gesture interaction system is set to 2; the available depth range is set to 0.5m-0.8m; the maximum number of the points in the hand contour located inside the circle is set to 25 and 50 respectively; the minimum radius of the palm is set to 33; the minimum distance of the candidate line (convex hull) is set to 50.

The original algorithm easily makes mistake in estimating the center of the palm, when a large part of the forearm enters the available depth range, as shown in Fig. 5. So the experiment focuses on the condition - a large part of the forearm enters the available depth range. Under this condition the user is required to make three hand postures: open the hand, extend two fingers and make a fist. In order to test the result of the wrist recognition algorithm, the experiment captures three pictures from the real gesture interaction system and compares the results of using the wrist recognition algorithm with the results of not using this algorithm, as shown in Fig. 6. The wrist recognition algorithm can achieve the desired result, isolating the hand from the forearm to avoid making mistake in estimating the center of the palm, although the recognition is not very precise.

In order to compare the new algorithm of estimating the center of the palm with the original algorithm, the experiment captures three groups of data, based on the number of the points inside the hand contour. The first group is shown in Fig. 7-a, 7-b and 7-c. The numbers of the points in the hand contour are 625, 470 and 505, respectively; the numbers of the points inside the hand contour are 7085, 5919 and 7335, respectively. The second group is shown in Fig. 7-d, 7-e and 7-f, the numbers of the points in the hand contour are 1074, 1356 and 1155, respectively; the numbers of the points inside the hand contour are 11205, 12362 and 13047, respectively. The third group is shown in Fig. 7-g, 7-h and 7-i, the numbers of the points in the hand contour are 1407, 1478 and 1024, respectively; the numbers of the points inside the hand contour are 16716, 17961 and 18990, respectively.

For each picture the new algorithm of estimating the center of the palm is tested 10 times, and the average of the results is calculated. Table I contains the results of using the new algorithm of estimating the center of the palm.

It is clear from Table I that the new algorithm of estimating the center of the palm has high efficiency, supporting real-time interaction. Especially in the conditions such as Fig. 7-a, 7-b, 7-c and 7-i, the new algorithm can rapidly find the center of the palm. That means the computing time of this algorithm does not correlate with the
number of the points inside the hand contour but correlate with the hand contour itself. The smoother the hand contour is, the higher the efficiency of the algorithm is. So when the user makes a fist, this algorithm will have the highest efficiency.

It can also be seen from Table I that appropriately relaxes the limitation, the number of the points in the hand contour located inside the circle, which can reduce the computing time. Moreover, the standard deviation of the coordinate values shows that the estimated center of the palm remains basically stable. So this algorithm can balance the speed and the precision based on the actual demand.

Table II shows the comparison among the original algorithm, the improved original algorithm and the new algorithm. The improved original algorithm refers to: as the original algorithm can’t support real-time interaction, the improved original algorithm does not take every point in hand contour and inside hand contour, but takes every 8th point. The new algorithm here is a bit different from the new algorithm proposed before. The new algorithm here also takes every 8th point in hand contour. It can be seen from the comparison that the new algorithm here has a distinct advantage in the performance.

### Table I. The Results of Using the New Algorithm of Estimating the Center of the Palm

<table>
<thead>
<tr>
<th>Picture Number</th>
<th>Threshold Inside</th>
<th>Average computing time (sec)</th>
<th>Average number of the points in hand contour located inside the circle</th>
<th>Standard deviation of X of the center of the palm (pixel)</th>
<th>Standard deviation of Y of the center of the palm (pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-a</td>
<td>25</td>
<td>0.016354</td>
<td>14.8</td>
<td>2.002776</td>
<td>6.963396</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.009256</td>
<td>35.9</td>
<td>2.806738</td>
<td>8.327331</td>
</tr>
<tr>
<td>7-b</td>
<td>25</td>
<td>0.023305</td>
<td>17.7</td>
<td>0.875595</td>
<td>1.75119</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.009176</td>
<td>32.9</td>
<td>1.286684</td>
<td>2.674987</td>
</tr>
<tr>
<td>7-c</td>
<td>25</td>
<td>0.020285</td>
<td>16.2</td>
<td>0.471405</td>
<td>0.948683</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.01028</td>
<td>32.2</td>
<td>1.549193</td>
<td>5.850926</td>
</tr>
<tr>
<td>7-d</td>
<td>25</td>
<td>0.069835</td>
<td>18.1</td>
<td>5.349974</td>
<td>7.501111</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.027444</td>
<td>32.1</td>
<td>5.719363</td>
<td>7.633988</td>
</tr>
<tr>
<td>7-e</td>
<td>25</td>
<td>0.152618</td>
<td>17.7</td>
<td>1.766981</td>
<td>3.018462</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.040003</td>
<td>34.4</td>
<td>2.780887</td>
<td>3.938415</td>
</tr>
<tr>
<td>7-f</td>
<td>25</td>
<td>0.069672</td>
<td>14.9</td>
<td>1.135292</td>
<td>6.733828</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.028626</td>
<td>38.2</td>
<td>1.159502</td>
<td>6.835366</td>
</tr>
<tr>
<td>7-g</td>
<td>25</td>
<td>0.085116</td>
<td>11.4</td>
<td>2.699794</td>
<td>7.211103</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.038471</td>
<td>33.9</td>
<td>2.875181</td>
<td>7.367345</td>
</tr>
<tr>
<td>7-h</td>
<td>25</td>
<td>0.085419</td>
<td>18.1</td>
<td>2.616189</td>
<td>8.643559</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.02523</td>
<td>36.7</td>
<td>3.620927</td>
<td>9.25454</td>
</tr>
<tr>
<td>7-i</td>
<td>25</td>
<td>0.024086</td>
<td>16.8</td>
<td>2.503331</td>
<td>3.50238</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.010914</td>
<td>36.5</td>
<td>2.945807</td>
<td>4.473378</td>
</tr>
</tbody>
</table>

### Table II. Comparison Among Algorithms

<table>
<thead>
<tr>
<th>Comparison Among Algorithms</th>
<th>Picture Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7-a</td>
</tr>
<tr>
<td>The computing time of the original algorithm (sec)</td>
<td>0.4325</td>
</tr>
<tr>
<td>The computing time of the improved original algorithm (sec)</td>
<td>0.0068</td>
</tr>
<tr>
<td>The computing time of the new algorithm (sec)</td>
<td>0.0002</td>
</tr>
</tbody>
</table>
VI. CONCLUSION AND FUTURE WORK

The paper proposes the wrist recognition algorithm to separate the hand from the forearm, and proposes a new algorithm of estimating the center of the palm to reduce the computing time. When a large part of the forearm enters the available depth range, the wrist recognition algorithm can recognize the wrist and separate the hand from the forearm well, which is necessary for correctly estimating the center of the palm and lowers the requirement of the Kinect sensor placement. The new algorithm of estimating the center of the palm has higher efficiency than the original algorithms, as the geometric features of an acute triangle and a maximum inscribed circle are applied into this algorithm. The two algorithms proposed in this paper do not use the Kinect Skeleton, so the user can sit pretty close to the computer, and interact with the computer using some gestures on finger level.

The system presented in this paper uses the OpenNI framework. However, when the system changes to use the Kinect SDK, the performance of the system decreases sharply. The future work will concentrate on how to improve the performance of this system using the Kinect SDK.

ACKNOWLEDGMENT

The authors would like to thank the National Natural Science Foundation of China (61173124) and the Scientific Research Fund of Zhejiang Education Department (Y2011120954) for supporting this research.

REFERENCES