Hand tracking and fingertip detection based on Kinect*

Xinyi Wang1,*, Wenxiao Ge1, a, Jinhu Li2, b

1 Bell Honor School, Nanjing University of Post and Telecommunication. Nanjing, Jiangsu, China
2 Faculty of Science and Engineering. University of Nottingham Ningbo China. Ningbo, Zhejiang, China
*Corresponding author e-mail: xy_wang98@163.com, a gewenxiao15ban@163.com, b zy22149@nottingham.edu.cn

Abstract. With the rapid development of computer technology and sensor performance, hand gesture recognition (HGR) has become one of the most promising research in human computer interaction field. Among all the procedures of HGR, real-time hand tracking is of great importance, which also faces many existing challenges. In this paper, an effective method was proposed for hand tracking in real time using Kinect, including the hand region extraction and fingertips determination. During the former procedure, we present a novel two-scan method based on connectivity analysis and recursion algorithm to extract the hand region in real time. After this, in order to determine the fingertips, we first implemented a fast hand convex hull extraction algorithm to obtain the primary candidate points. Then two typical constraints based on hand shape characteristics are applied to determine the final fingertip points. The subsequent experiments verified that our proposed strategy could significantly improve the quality of real-time hand tracking.

1. Introduction

With continuous progress of computer technology and sensor performance, many novel service and applications has been proposed and implemented such as virtual reality (VR), intelligent household, industrial control, etc. Among various types of new technologies and service, human computer interaction technology, which is developed to the human-centered natural interface interaction stage [1], could translate users’ motions into computer language to understand their intentions and further control computer operation. Compared with other interaction ways, non-contact human computer interaction based on hand gesture is more common and flexible to express and communicate, which makes hand gesture recognition (HGR) become one of the most promising research fields.

Generally, accurate HGR can be achieved by using the wearable equipment like gloves, sensors, wires and so on. However, these specialized devices definitely lead to extra cost and inconvenience. Meanwhile, due to the landmark appearance of Kinect released by Microsoft, the research of HGR has reached a new stage, namely computer vision based HGR method. Without using extra device, this cost of HGR techniques could be reduced a lot [2]-[3]. More importantly, cv-based HGR methods can provide users with a more convenient and nature interaction way [4], making it become as the mainstream of HGR research field in recent years.

For HGR technology, the geometric hand features, like finger direction, number of fingertips and hand contour, play a significant role in gesture identification. Therefore, the detection and extraction of the hand from the complex background has outstanding impacts on the effectiveness of human
computer interaction. Generally, skin-color, shape and depth pixel values are the most common information for hand region detection. Skin-color based method filter out the hand region pixels via analyzing different component in corresponding color space, such as RBG, HSV, YCrCb and YUV [5]-[6]. However, the results could be heavily affected by the ambient illumination that is always dynamically changing in hand movement. Shape-based approaches are to detect similar contour for template matching. Therefore, a reliable database is needed and there are much complex post-processings for accuracy. Compared with the former two methods, depth pixels-based approach has no common rules and mostly depends on the morphological or spatial conditions, which makes its results can be immune to environment illumination. More importantly, there is no need for complicated subsequent processing. In this paper, a novel and effective approach was proposed to extract the hand region based on two scans using the depth image data.

Fingertips play an important role in the real-time hand movement description and are often used as an event trigger for human-computer interaction based on hand. Convex hull and convex defect, curvature detection and template matching are often used for fingertip detection. Curvature detection-based methods usually use the hand contour information to analyze the edge feature information. The point that has large curvature will be likely to be judged as fingertips. Although this method has a relatively good performance in efficiency, it cannot determine the number of fingertips since it is impossible to judge the stretching, bending or self-occlusion of the hand fingers in the image, for which misidentification is easy to occur. The methods based on template matching adopt SVM [7] or PCA [8] to match the set of pixel points in the image to obtain the position of fingertips, with a low robustness to adapt dynamically changed scenarios. As for convex hull detection-based approach, the computation is simple so that this method is with a high real-time performance for fingertips detection. However, not all the convex hulls vertexes are fingertips. Therefore, a fast hand convex hull extraction algorithm was implemented to obtain the primary candidate points, then two typical constraints based on hand shape characteristics are used for determining the final fingertip points.

This paper presents a real-time hand tracking strategy based on depth image data from Kinect, including hand region extraction and accurate fingertips determination. The rest part of this paper is organized as follows: Section II introduces the proposed approaches in detail for real-time hand tracking in each procedure. Then section III analyzes the performance of our method through reliable experimental results. The last section draws the conclusions.

2. Methodology

2.1. Data Preprocessing

2.1.1. Occlusion Repair. Kinect could detect 25 human body joints of 6 people at a time (illustrated in Fig.1) and provide corresponding original RGB or depth data information. In general, the data provided by Kinect is accurate and stable with low noise. However, the occlusion of tracked joints might make the real-time data jitter sharply, reducing the data reliability. In this case, we should repair the occluded joints considering the length invariance of human skeleton and the continuity of hand gesture movement. This process is shown in detail as follows:
Before processing the skeleton information, a standard human skeleton information database was established to store the standard human skeleton length. After got the human skeleton information, it was assumed that the human spine skeleton information was reliable. Considering the length of the spine and the length of the spine in the database to be $L^0$ and $L_{base}^0$, a proportional factor was defined as $r$:

$$r = \frac{L^0}{L_{base}^0} \quad (1)$$

From formula 1 it could be got the ideal skeleton length of human body:

$$L' = r \times L_{base}^i \quad (2)$$

where $L'$, $L_{base}^i$ represent the ideal skeleton length of the target and the skeleton length in the standard database, respectively.

According to ideal length of the skeleton, the reliability of the joint could be preliminarily determined. If the coordinates of pixel A and B in frame $k$ are $(x_A, y_A, z_A)$ and $(x_B, y_B, z_B)$, the length of skeleton $L^i_{AB}$ could be calculated from the Euclidean distance between the two points. Consider the acceptable error is $\varepsilon$. If this length $L^i_{AB}$ is within $(L^i_{AB} - \varepsilon, L^i_{AB} + \varepsilon)$, we considered point B to be preliminarily credible.

Step2:

Since the motion of human joints is continuous, the velocity of the joints in a time interval should also be within a certain range. The based on time interval between adjacent frames as $t$, the coordinates of joint B in frame $k-1$ and frame $k$ are respectively expressed as $(x_{B, k-1}, y_{B, k-1}, z_{B, k-1})$ and $(x_{B, k}, y_{B, k}, z_{B, k})$, then we could get the velocity in three axis directions of joint B on frame $k$ as follows:

$$\begin{align*}
\dot{v}_{x:B} &= \frac{x_{B, k} - x_{B, k-1}}{t} \\
\dot{v}_{y:B} &= \frac{y_{B, k} - y_{B, k-1}}{t} \\
\dot{v}_{z:B} &= \frac{z_{B, k} - z_{B, k-1}}{t}
\end{align*} \quad (3)
$$

According to the information of the first five frames, the velocities in three axis directions of the $k-3$, $k-2$, $k-1$ and $k$ frames at point B could be calculated respectively. In consideration of the previous
speeds and the acceptable error, the speed range of joint B in frame k can be determined. If the actual velocity is in this range, then joint B can be trusted.

Step3:
For the joints that cannot be trusted (missing or partly occluded), their information will be repaired or modified according to skeleton length and mannequin constraints. The principles are the same as what have discussed above.

2.1.2. Kalman Filtering
Environment changes and noise will make Kinect data jitter or even cause data loss. In order to make data more stable and reliable, a Kalman filtering algorithm was adopted to further process the joint data provided by Kinect after occlusion repair.

After reading and repairing the data by Kinect, the current location information was obtained. By kinematics analysis, we got the speed information from multiple frames of data, as well as the control information. \( T_k \) describes the relationship between the target variable and the variable read by Kinect. The covariance matrix between variables and gaussian deviation in the prediction process were respectively denoted as \( Q_k \) and \( G_k \). Then we could get the Kalman gain:

\[
K_{\text{gain}} = T_k \cdot Q_k \cdot T_k^T (T_k \cdot Q_k \cdot T_k^T + G_k)^{-1}
\]  

After occlusion repair and Kalman filtering, a series of stable and reliable data including the tracked palm center point was got, to further extract the hand region.

2.2. Hand Region Extraction
Traditional processing methods for two-dimensional depth images will cause interference to the extraction of hand region and affect the authenticity of three-dimensional information. Considering that the hand changes in real time during movement and the target area is relatively concentrated, this paper presents a connectivity analysis method based on two scans to completely extract the hand region from the complex background in real time.

Step1: The First Scan
During the first scan, the palm center point obtained after Kalman filtering is taken as the search center. All pixel points are preliminarily filtered to retain as many pixels as possible close to the depth value of the palm center. It should be pointed out that in order to avoid missing useful information, 1.2 times of the largest skeleton length between joints in the corrected hand region is taken as the search radius. Though the first scan retains some useless and distracting information, the second scan will solve this problem.

Step2: The Second Scan
Based on the depth image data, we use a connectivity analysis method to further extract the exact hand region. Due to the fact that the traditional connectivity analysis algorithms need to traverse all the candidate pixels twice, which are not efficient for the real-time hand region extraction, the second scan adopts a recursion algorithm to analyze the connectivity of the pixels obtained in the first scan. That is, if the difference between the depth value of a certain pixel point and the depth value of its surrounding 8 pixels is less than a certain threshold, the two points are considered connected. Then we take this pixel as the center of the search and compare it again with its surrounding 8 pixels. Go on like this, all the pixels in the hand region can be searched layer by layer illustrated in Fig.2. And the flow of this algorithm is shown in detail in Fig.3.
2.3. Finger Detection

2.3.1. Convex Hull. Suppose there is a set of points on a certain surface and connect the least few points into a convex polygon which contains all the points in the set (all points in the plane are inside or on the convex polygon). This convex polygon is called the two-dimensional convex hull of this point set. For the point set of the extracted hand region pixel, it could be found that its convex hull vertexes include the fingertip points, so via hand convex hull extraction, we can choose some fingertip candidate points, and then the final fingertip points can be determined by some geometry constraint conditions.

Hand convex hull extraction:
To extract the convex hull of the hand region, the commonly used way is the Graham scanning method. Due to this algorithm need to scan and sort all the pixel points, it takes a long time to obtain the convex hull of the hand region. For the purpose of determining the fingertips in real time, we propose a fast-convex hull extraction algorithm that gradually eliminate the points inside the convex hull by using the recursive method. The specific algorithm process is shown in table 1 as follows:

| Table 1. Hand convex hull extraction algorithm |
|-----------------------------------------------|
| Input: pixels in hand region                  |
| Output: hand convex hull vertices (candidate fingertip points) |
| 1. Scan all the pixel points to determine the two extreme points A and B which divide the entire hand convex hull into the upper and the lower part via coordinate judgment, showing in Fig.4. |
| 2. Gradually eliminate the points inside the two divided convex hulls respectively by recursion process: |
| Find the point C farthest from the line segment AB, connect AC and BC. |
| Set A, C recursively as the two extreme points, redo 1. |
| Until the convex hull that correspond to certain two extreme points cannot be found. |
| The farthest points are the convex hull vertices we want, showing in the Fig. 4. |

2.3.2. Fingertip Determination. From Fig.5 (a), we can see that not each convex hull vertex is a fingertip point, so we need to further filter out the real fingertips from these candidate points.

As shown in Fig.5 (b), $p_{ij}$ is a certain point on the hand contour segment between the two adjacent convex hull vertices, and it has the furthest distance to the line segment from $r_i$ to $r_j$. We call $p_{ij}$ the convex defect point. And the furthest distance (expressed as $d_{ij}$) is defined as the convex defect depth of these two points. Based on the hand shape characteristics, we can find that the convex hull defect depth between two fingertips is much greater than other adjacent convex hull vertices, so we can define a threshold $d$ as a constraint for fingertip determination, which can be expressed as follows:

$$d_{ij} \geq d$$
That is, as long as the defect depth $d_{ij}$ (between $r_i$ to $r_j$) is greater than $d$. $r_i$ and $r_j$ are the possible fingertip points.

Besides, as we have tracked the accurate hand palm point using Kinect before, it can be considered to create another constraint for fingertip determination. From Fig. 5 (b), it can be seen that the distance from the fingertip to hand palm AC is greater than the distance from the convex defect point to hand palm BC, but for other vertices, the distance AC is almost equal to BC, so the second fingertip constraint could be expressed as follows:

$$MIN(BC, AC) / MAX(BC, AC) \leq 0.8$$ \hspace{1cm} (6)

To summary, from the two constraint we discussed above, we can effectively determine the final fingertip points.

3. Experiment/Performance Analysis

In this section, some experiments were carried out as follows, for the purpose of verifying the reliability and accuracy of our proposed method. And the experiment results show that our approach is of great effectiveness for hand region extraction and further fingertip determination.

3.1. Hand Region Extraction

In this part, the effects of proposed two scans method based on connectivity analysis and recursion algorithm in hand region extraction are evaluated. In order to verify the efficiency and reliability of our approach, we carry out the hand extraction procedure experiment via our method and the traditional connectivity analysis algorithm (need to traverse all the pixels in the candidate region), respectively. This process is repeated 100 times for average and the final results are listed as follows:

Fig. 6 (a) illustrates the hand region extracted via traditional connectivity analysis algorithm, while Fig 6 (b) and Fig 6 (c) shows the hand region after the first scan and the second scan of our method, respectively.

From these three images, we can see that after the first scan of our approach, there exists some interference points, but after the second scan, the extra points are almost removed, and the final extracted hand region is almost the same as the result obtained by traditional method.

![Fig.6. Comparison diagram of hand region extraction results](image_url)

We also compute the number of pixels that these two algorithms need to process, respectively, which is shown in Table 2. From the table, we can find that the traditional connectivity analysis algorithm needs to process about 237440 pixels on average for the hand gesture shown above, while our approach only needs to process about 1782 pixels. And the time spent by our method (0.000439 second on average) is much shorter than traditional way (0.014864 second on average). These results prove that our approach outperforms the traditional connectivity analysis way in efficiency for hand region extraction.

| Parameters          | Traditional continuous region analysis algorithm | Our algorithm |
|---------------------|--------------------------------------------------|---------------|
| Target area pixels  | 237440                                           | 1782          |
| Iterations          | 0.014864                                         | 0.000439      |

Table 2. Algorithm efficiency comparison
3.2. Performance Analysis of Fingertips Determination Strategy

3.2.1. Algorithm accuracy test. We illustrate effects of our fingertips determination approach in two aspects: accuracy and robustness. The process is demonstrated as follows:

Performance in accuracy:
Considering that the hand gestures representing numbers 1-5 are much commonly used in daily life, we select these gestures to carry out the first experiment, for the purpose of verifying our fingertips determination strategy’s performance in accuracy. In order to eliminate the influence of different hand shapes, three different people are invited to perform this experiment. Each person make 5 gestures of representing number 1 to 5 and each gesture is repeated for 40 times. Finally, we collect 600 frame images in all, and the statistic results are shown in table.

| Fingertip | Image number | Error number | Accuracy  |
|-----------|--------------|--------------|-----------|
| 1         | 120          | 2            | 100%      |
| 2         | 120          | 1            | 100%      |
| 3         | 120          | 2            | 98.83%    |
| 4         | 120          | 2            | 98.83%    |
| 5         | 120          | 0            | 98.83%    |

The number of fingertips is corresponding with the number that the hand gesture represents. If these two figures are not equal, the corresponding frame is not detected correctly for fingertips determination. From table 3, we can see that the error frame is few and the detection accuracy of our strategy reaches 98.83%, representing very accurate fingertips positioning.

3.2.2. Performance in robustness. In the previous experiments, all the gestures are faced with Kinect. Considering that in many scenarios, the hand can be back to the Kinect, so in order to verify whether our strategy’s accuracy can be influenced by this factor, we further carry out experiments using the gesture representing number 5, and record the results for condition that the hand is face or be back to the Kinect, respectively, as shown in Fig.7. This process is repeated for 50 times and the results are listed in table 4.

| Gesture | Image | Error | Accuracy |
|---------|-------|-------|----------|
| Positive| 50    | 1     | 98.00%   |
| Negative| 50    | 0     | 100%     |

(a)
From the table 4, it could be found that the condition for hand facing or being back to the Kinect has less influence on the accuracy of fingertips detection. However, during the experiment process, we find the detection accuracy has a great relationship with the distance from hand to Kinect. When the hand is far away from Kinect, it can be seen that the adjacent fingertip points have the tendency to coincide (illustrated in Fig.8) and the detection accuracy will be correspondingly reduced.

Fig. 8. Fingertip detection at a long distance

4. Conclusion
In this paper we proposed an effective method to extract the hand region and detect the fingertips using Kinect sensor in real-time. Based on the hand palm point captured by Kinect, target hand region could be roughly extracted in the first scan. In order to overcome the side effects on Kinect joints data due to the environment and illumination changes, occlusion repair and Kalman Filter Algorithm was used to modify the tracked hand palm point position at first. After obtaining the rough hand region, a novel connectivity analysis method based on recursive algorithm is applied to refine the hand region pixels in real time during the second scan. For further fingertips determination, the primary candidate points are obtained by a fast hand convex hull extraction algorithm, then two typical constraints based on hand shape characteristics were implemented to determine the final fingertip points. The subsequent experiments verify that our proposed strategy is of great effectiveness to improve the performance for hand tracking in efficiency and accuracy.

Acknowledgments
This work was supported by the National Science Technology Innovation Training Program and was carried out under grants provided by Nanjing University of Post and Telecommunications. We really appreciate our university and tutor giving us so much help and encouragement.

References
[1] Sigal L, Sclaroff S, Athitsos V. Skin color-based video segmentation under time-varying illumination[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(7):0-877.
[2] Baudel T, Beaudouin-Lafon M. Charade: remote control of objects using free-hand gestures[J]. Communications of the ACM, 1993, 36(7):28-35.
[3] Sturman, D. J., & Zeltzer, D. A survey of glove-based input[C]. IEEE Computer graphics and Applications, 1994, 14(1), 30-39.
[4] Zhang Z. Microsoft Kinect Sensor and Its Effect[J]. IEEE Multimedia, 2012, 19(2):4-10. Conference Name: ACM Woodstock conference

[5] Wang, Y. R., Lin, W. H., & Yang, L. A novel real time hand detection based on skin-color[C]. In Consumer Electronics, 2013, 141-142.

[6] Tang H T H, Feng Z F Z. Hand's Skin Detection Based on Ellipse Clustering[C]// International Symposium on Computer Science & Computational Technology. IEEE, 2008.

[7] Yoon, S., Feng, J., & Jain, A. K. Altered fingerprints: Analysis and detection[C]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(3), 451-464.

[8] Dardas, N. H., & Georganas, N. D. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques[C]. IEEE Transactions on Instrumentation and measurement, 2011, 60(11), 3592-3607.