Research Article

Kinect Sensor-Based Long-Distance Hand Gesture Recognition and Fingertip Detection with Depth Information

Xuhong Ma and Jinzhu Peng (ORCID)

School of Electrical Engineering, Zhengzhou University, Zhengzhou, Henan 450001, China

Correspondence should be addressed to Jinzhu Peng; jzpeng@zzu.edu.cn

Gesture recognition is an important part of human-robot interaction. In order to achieve fast and stable gesture recognition in real time without distance restrictions, this paper presents an improved threshold segmentation method. The improved method combines the depth information and color information of a target scene with hand position by the spatial hierarchical scanning method; the ROI in the scene is thus extracted by the local neighbor method. In this way, the hand can be identified quickly and accurately in complex scenes and different distances. Furthermore, the convex hull detection algorithm is used to identify the positioning of fingertips in ROI, so that the fingertips can be identified and located accurately. The experimental results show that the hand position can be obtained quickly and accurately in the complex background by using the improved method, the real-time recognition distance interval can be reached by 0.5 m to 2.0 m, and the fingertip detection rates can be reached 98.5% in average. Moreover, the gesture recognition rates are more than 96% by the convex hull detection algorithm. It can be thus concluded that the proposed method achieves good performance of hand detection and positioning at different distances.

1. Introduction

Nowadays, the interaction between people and the machines is mainly completed through the mouse, keyboard, remote control, touch screen, and other direct contact manner, while the communication between people is basically achieved through more natural and intuitive noncontact manner, such as sound and physical movements. The communication by natural and intuitive noncontact manner is usually considered to be flexible and efficient; many researchers have thus tried efforts to make the machine identify other's intentions and information through the noncontact manner like people, such as sound [1], facial expressions [2], physical movements [3], and gestures [4, 5]. Among them, gesture is the most important part of human language, and its development affects the nature and flexibility of human-robot interaction [6–10].

In the past decades, gestures were usually identified and judged by wearing data gloves [11] to obtain the angles and positions of each joint in the gesture. However, it is difficult to use widely due to the cost and inconvenience of wearing the sensor. In contrast, the noncontact visual inspection methods have the advantage of low cost and comfort for the human body, which are the currently popular gesture recognition methods. Chakraborty et al. [12] and Song et al. [13] proposed the skin color models utilizing image pixel distribution in a given color space, which can significantly improve the detection accuracy in the presence of varying illumination conditions. However, it was difficult to achieve the desired results using the model-based methods because of the light sensitivity during the imaging process. The algorithm-based noncontact visual inspection methods were also used to conduct the gesture recognition, such as the hidden Markov model [14], the particle filter [15], and Heer features Ada-Boost learning algorithm [16]; however, it is difficult to execute real time due to the complicated algorithms. The above results cannot acquire gestures efficiently in real time since only the insufficient 2D image information was used.

Therefore, it is inevitable that gesture recognition by 2D image is replaced by 3D with depth information. In general, 3D information can be acquired by binocular cameras [17], Kinect sensor [18–20], Leap Motion [21], and other devices. Those devices can be usually utilized to obtain depth

information by spatial relationship of different direction [17] or infrared reflection [22], which can conveniently acquire noncontact image for recognition and classification instead of wearing the complicated equipment. For example, Ťupa et al. [23, 24] presented the detection of selected gait attributes by Microsoft Kinect image and depth sensors to track movements in three-dimensional space. Youness et al. [25] proposed a real-time human pose classification technique by using skeleton data from a depth sensor. However, the calibration process of binocular camera is usually complex, and the recognition distance of Leap Motion is only from 2.5 cm to 60 cm. Due to the noncalibration and long-distance recognition, Kinect sensors have been widely used in body pose detection [26, 27], skeleton tracking technology [28], and other aspects.

Gesture extraction from the complex image is considered to be important, and more information generally leads to more accuracy and larger scope of the gesture recognition. Kinect sensor is usually selected to acquire gestures to obtain extra depth information, and it has demonstrated a significant performance of light sensitivity in gesture recognition. However, there are still unsolved issues of gesture finding, and segmenting with depth information needs to be further treated. Recently, researchers have focused on the recognition problem instead of the problem of gesture segmentation in the applications of hand gesture recognition. The gesture segmentation methods were usually utilized by the direct distance interval setting [4, 5, 29] or the hand being the frontmost object [30]. The simplified methods demonstrated to be quick and effective; however, the distance between the hand and Kinect sensor is restricted so that the hand gestures can only be recognized by moving the hand to a specific position and keeping the distance during the whole process. In general, the researchers tried their best to seek the human-robot interactions in a natural way like the communication of human beings. However, the restriction of a shortage of distance is unnatural in the above and existing literatures. The gesture recognition methods such as template matching [31] or finite-state machines [32] were usually used, where high classification rates could also be obtained. However, only specific gestures can be recognized by the above methods. The convex hull detection algorithm [33] recognizes gestures with the finger hull and can identify each fingertip position of the human hand. It can get more gesture information and have potential advantage. The convex hull detection algorithm was utilized to recognize gesture by the finger hull, and each fingertip of the human hand was thus positioned.

In this paper, an improved threshold segmentation method is proposed based on the Kinect sensor with the depth information for long-distance recognition. The proposed method not only has the advantage of light insensitivity but it also can extract the gestures accurately in a wide range of contexts and complex backgrounds where the hand gesture is not covered completely by the front objects. Firstly, the RGB image data and the depth image data obtained by the Kinect sensor are preprocessed by median filtering. Secondly, combined with the depth of information and skin color threshold, an improved spatial stratification method is proposed to extract gestures; the gestures can be thus identified within wide contexts in complex backgrounds. Finally, the local neighbor method is conducted to segment the ROI of the human hand. In order to verify the efficiency of the proposed method, the k-cosine curvature method is also presented to detect the fingertips and recognize the number gestures. The experimental results demonstrate that the proposed method can achieve a good performance and has strong robustness.

The remainder of this paper is organized as follows. In Section 2, hand gestures are extracted from an image by the improved threshold segmentation method. In Section 3, the algorithm of fingertip detection is described. In Section 4, the experimental tests are conducted to detect the fingertips and recognize number hand gestures. And conclusions are drawn in Section 5.

2. Gesture Extraction

Gesture extraction includes preprocessing and hand segmentation. Preprocessing is to register the depth image and RGB image and to process the spatial stratification of depth distance. Hand segmentation is to extract the hand area at different distances from the complex background.

2.1. Hand Recognition Preprocessing. In order to carry out the hand segmentation, it is necessary to register the depth image and RGB image, to stratify the depth distance, and to filter the depth image.

The first step is to register the depth image and RGB image. The resolution of RGB image obtained by the Kinect sensor is 1920×1080, while the resolution of depth image is 512×424 which is converted by depth information. Moreover, the RGB camera and IR camera of the Kinect sensor are at different points. Therefore, it is mismatched in the spatial size and cameras positioned between the RGB image and the depth image. In order to obtain the corresponding RGB information and depth information for each point, the point on the depth image has to be registered to match a corresponding point on the RGB image. In this paper, CoordinateMapper in Kinect v2 API is used to register the RGB image and depth image. The result is basically accurate except the burr on the edge of the object due to the distance gap. However, it can be ignored because the accuracy can meet the requirement in the following calculations. After registration, assume that the target image is P, which is composed of pixel sets $P = (p_1, p_2, \ldots, p_n)$. The value of each p_i is $p_i = (x_i, y_i, d_i, r_i, g_i, b_i)$, where (x_i, y_i) represents the pixel coordinate position of p_i, d_i represents the depth value of p_i, and $(r_i g_i b_i)$ represents the RGB element value of p_i.

The second step is to spatially stratify the depth distance. The aim of the spatial stratification is to seek the characteristics of the human hand in depth space from near to far, so that one can avoid the limitation of traditional methods where only gestures in the forefront can be detected. According to our experimental investigation, the detection range of the depth camera is 0.5 m~4.5 m; however, it is not sufficient to identify the gestures when the distance between the Kinect sensor and the human hand is greater than 2.5 m. Therefore,

the stratification range of our experimental test is selected to be 0.5 m~2.0 m. Taking into account the real time and accuracy of the Kinect sensor, we set 0.1 m to be the layer step; therefore, P can be divided into 15 parts, and the kth layer is represented by P_k pixel set, then

$$0.5 + 0.1(k-1) < d_i \leq 0.5 + 0.1k. \tag{1}$$

The last step is to filter the depth image. Since the depth image is obtained by calculating the random speckles, which are produced by infrared light of IR camera reflected on a rough object, the points without values or nonuniform regions would be thus inevitable. The median filtering method is a nonlinear smoothing technique, which sets the gray value of each pixel to the median of the gray values of all the pixels within a certain neighborhood of the point. In comparison to other filters such as mean filter and Gaussian filter, the main advantage of median filtering method is that the isolated noise points can be eliminated efficiently while the gesture edge information can be well retained. Therefore, in this paper, the depth image is preprocessed by using the median filter to get rid of the small noise points in the image, where the aperture linear size of median filtering is set to be 3. The medianBlur [34] in the OpenCV is used for depth image filtering due to its fast and effective performance.

2.2. Hand Segmentation. After registering the depth image and RGB image, P is detected layer by layer by spatial stratification. There are two steps to obtain the ROI of gesture. The first step is to find the approximate location of the gesture and distinguish the right or left hand and other uncovered parts of the body by using GetJoints in the Kinect v2 API. The second step is to detect skin color layer by layer in the approximate image frame. In this way, the accurate ROI of gesture can be then determined. In this paper, the object detection is conducted from the object near the camera, which is combined with the depth information; the target can be thus located quickly when the conditions of RGB information are met. In this way, the ROI area of the target depth image can be obtained. From the first layer space P_1, the hand detection would be turned to the next layer to detect until the RGB value of p_i in P_k point set in the range of the skin color values. Then, the area where the hand is located would be determined if r_i, g_i, b_i in $p_i = (x_i, y_i, d_i, r_i, g_i, b_i)$ satisfy

$$(r_{\min} < r_i \leq r_{\max}) \wedge (g_{\min} < g_i \leq g_{\max}) \wedge (b_{\min} < b_i \leq b_{\max}), \tag{2}$$

where $r_{\min}, g_{\min}, b_{\min}$ and $r_{\max}, g_{\max}, b_{\max}$ are the skin color of the lower and upper bounded values of the tricolor, respectively.

When the number of interest points meets the condition that is a given certain number of points in this paper, one can rule out the possibility of interference noise or other uncertainties. Then, these points are determined in the kth layer and to be the skin color. In this way, the area where the hand is located can be finally determined.

After determining the hand position, the local neighbor method is used to obtain the ROI area in this paper; that is, a square is chosen to sign the hand position, where the side length is b and the interest point is taken as the center. Due to the large interval range from 0.5 m to 2.0 m between the
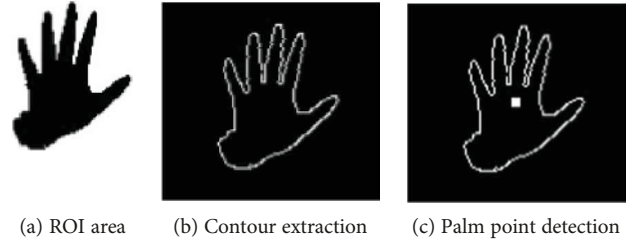

(a) ROI area   (b) Contour extraction   (c) Palm point detection
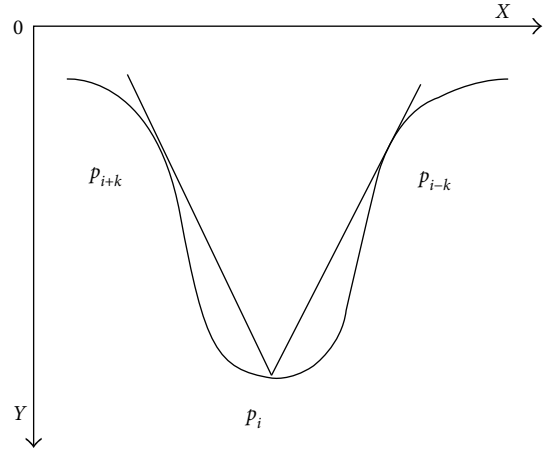
FIGURE 1: Hand detection process.

FIGURE 2: k-cosine curvature algorithm.

hand position and Kinect sensor, the hand pixels in the image appear to be big when near and small when far.

Remark 1. The appropriate value of b in a short distance of ROI area would be too large in a long distance, and the calculation accuracy and speed performance would thus be decreased. On the contrary, the necessary information would be lost and the location may not be obtained. Therefore, the value of b should be chosen according to the layer size k to ensure the accuracy of the hand segmentation.

According to the above analysis, the ROI area can be obtained, which is shown in Figure 1(a). In the ROI area, the gray value of the background in the image is set to be 255, while the gray value of the gesture area is set to be 0, the binary image of the hand gesture can be thus obtained. Let the gray scale value set of ROI be $P_{ROI} = (p_1, p_2, \ldots, p_m)$; therefore, P_{ROI} is the pixels set of the gesture location.

3. Fingertip Detection

After extracting the binary image of the hand gesture from the complex background, contour profile is extracted from the ROI region to find the hand, the palm point is then detected, and fingertips point can be finally positioned based on the obtained contours and palm point.

3.1. Hand Contour Extraction. In this paper, the FindContours algorithm [34] is utilized to extract hand gesture
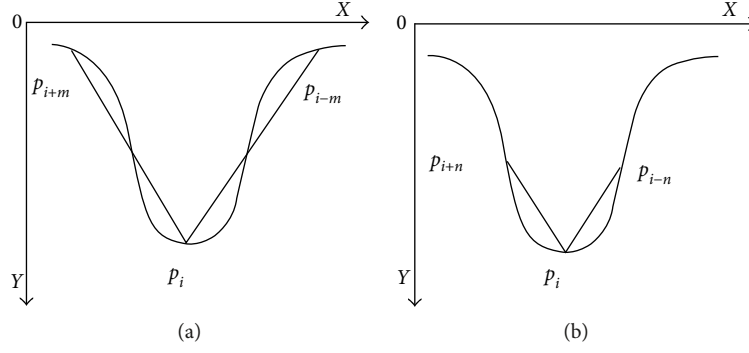
(a)   (b)

FIGURE 3: Feasibility when k values is m(a) and n(b).

contour from the ROI region. Contour point extraction is generally achieved by comparing the size between adjacent pixels. The basic principle of the FindContours algorithm is to find the contour by detecting the boundary between black and white regions of the binary image. In the previous section, we have obtained the binary image of hand gesture, the Find-Contours algorithm is then used to extract the gesture contour from the ROI region, which is shown in Figure 1(b).

After obtaining the gesture contour, the contour points are stored clockwise by using the array $P_{\text{Seq}} = (p_1, p_2, \ldots, p_i, \ldots, p_n)$.

3.2. Palm Point Detection. Kinect sensor provides the palm detection of hand gestures, where the end of the upper limb is considered to be the palm point based on the detection and recognition of human skeleton. However, this method can only be used in the case where the whole body image can be detected. Moreover, the hand detection provided from Kinect sensor has errors since it is usually in the inferred state. Therefore, the gesture contour center is used as the palm point in this paper. According to our above experiment test, the hand contour of the hand gesture in the ROI has been obtained. The center point of the contour is calculated as the palm point O based on the coordinate array of hand contour, which is shown in Figure 1(c).

Remark 2. Although the above palm point detection method has low accuracy, it is relatively fast and stable. In this paper, the palm coordinate point is mainly detected for excluding the nonfingertip groove point as the fingertip point is calculated; the requirement of precision is thus not too high. Therefore, the method can meet the requirements of calculation.

3.3. Fingertip Calculation. To track the characteristics of hand gestures, it is necessary to know the feature points of the hand in the human-machine interaction. The most important feature of human hand is the fingers; therefore, we have to locate finger positioning, that is, to find the fingertip points. In the previous subsection, we obtained the gesture contour image, where the main feature of the fingertips is convex hull. Therefore, in this paper, the k-cosine curvature algorithm, which is shown in Figure 2, is used to calculate the curvature values of the gesture contours. The points, which are obtained by the reasonable setting of the
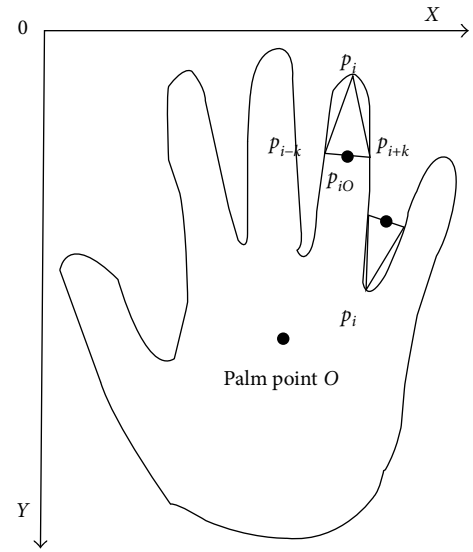

FIGURE 4: Convex point at the fingertips and groove point.

parameters and matched curvature values, are the coordinates of the fingertips.

In Figure 2, p_{i+k} is the succedent kth point of p_i in the clockwise contour array, and p_{i-k} is the previous kth point.

Define the two vectors formed by p_i, p_{i+k}, and p_{i-k} on the gesture contour curve as

$$\boldsymbol{\alpha}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k}),$$
$$\boldsymbol{\beta}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k}). \tag{3}$$

Then, the k-cosine value of p_i can be obtained as

$$e_{ik} = \frac{\boldsymbol{\alpha}_{ik} \cdot \boldsymbol{\beta}_{ik}}{|\boldsymbol{\alpha}_{ik}||\boldsymbol{\beta}_{ik}|}. \tag{4}$$

In general, the appropriate interval of e_{ik} needs to be selected. A point that complies with the interval can be considered to be the required corner.

Remark 3. It is necessary to select an appropriate k value so that the position of each fingertip can be precisely detected. In order to handle the inaccuracy problem caused by the difference in size of the gesture at different distances, and to
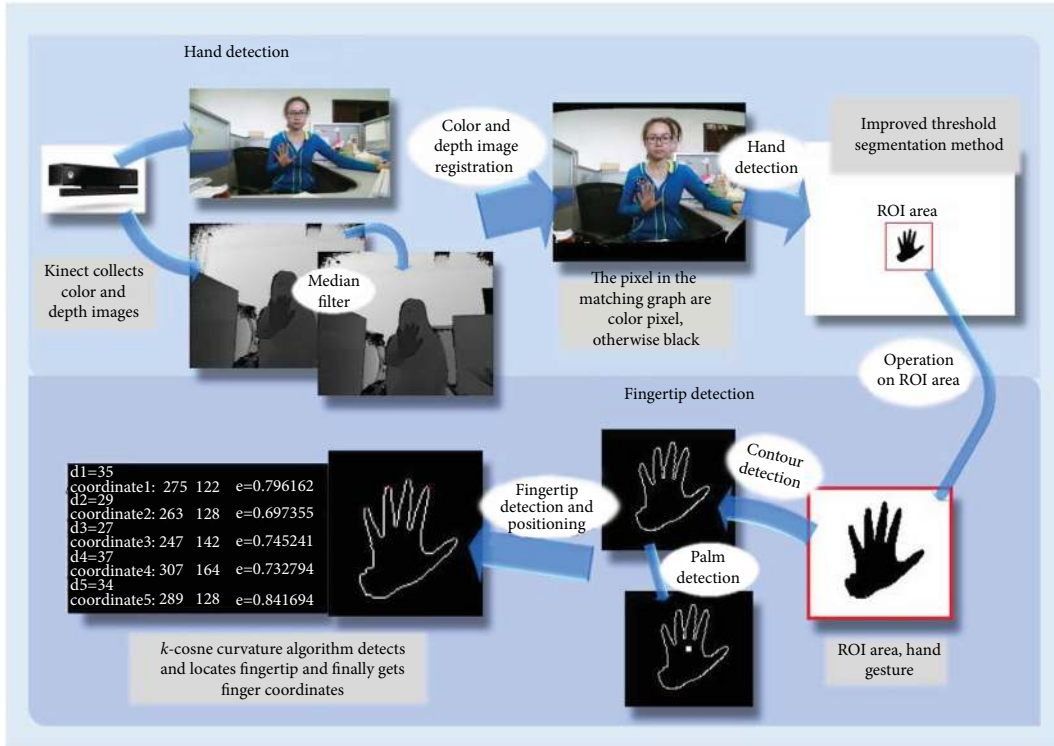
FIGURE 5: Experimental process.

prevent the burr problem of edge curve caused by unsmooth contour, the range of k value should be selected carefully.

In this paper, we select $k \in (n, m]$, the k-cosine of p_i, as $k = n, n + 1, \ldots, m$; $e_{in}, e_{i,n+1}, \ldots, e_{im}$ can be then calculated, as shown in Figure 3. Then the maximum cosine value e_{ik} can be found when k is a certain value between m and n, which is exactly the k-cosine value of p_i.

Based on the k-cosine of contour point, the angle ρ_i between $\boldsymbol{\alpha}_{ik}$ and $\boldsymbol{\beta}_{ik}$ can be calculated. Given an angle ρ_0, the corresponding contour point p_i is suspected as the finger point when $\rho_i < \rho_0$.

As shown in Figure 4, the contour point that satisfies the condition $\rho_i < \rho_0$ can be the convex point of the fingertip or the groove point between fingers. Therefore, the distance between this point and the palm point is utilized to further determine the fingertip or finger groove point. In Figure 4, assuming that p_{i0} is the midpoint of p_{i-k} and p_{i+k}, and the distance between p_{i0} and palm point O is d_{i1}, the distance between p_i and palm point O is d_{i2}. That distance is the length between two points in the Cartesian coordinate system according to these point coordinates. Then

(1) p_i is the fingertip, if $d_{i1} < d_{i2}$.

(2) p_i is the groove point, if $d_{i1} > d_{i2}$.

4. Experimental Results and Discussion

In this paper, Microsoft Kinect 2.0 is used as the data acquisition device, and dynamic per frame rate is used for video
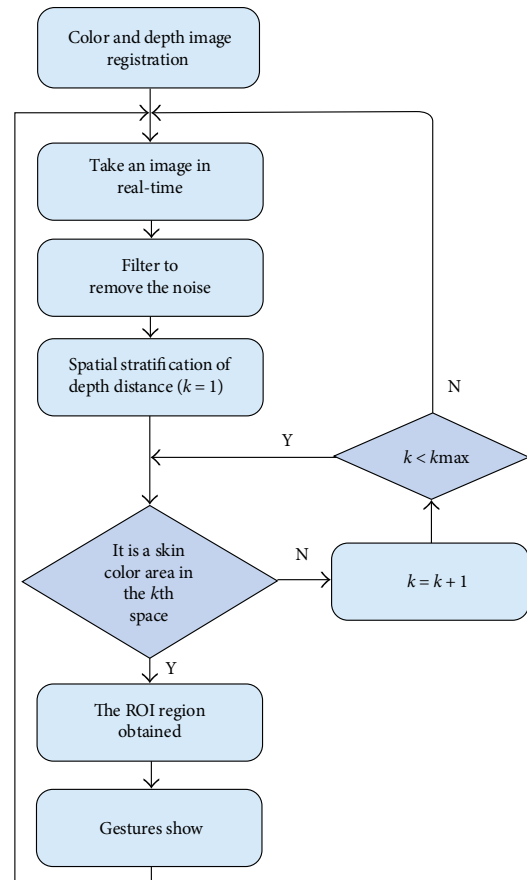

FIGURE 6: Flow chart of hand recognition.

(a) Distance is 0.6 m   (b) Distance is 1.0 m

(c) Distance is 1.5 m   (d) Distance is 2.0 m

Figure 7: Results of hand recognition.

data acquisition; the experimental tests are conducted by the Visual Studio 2010 platform using the C++ program. OpenCV is used for image processing, such as image data storing and contour points seeking. According to the above analysis, there are two steps in the experimental tests. First, the improved threshold segmentation method is used for hand recognition. Then, the fingertip convex hull detection algorithm is used for fingertip positioning and gesture recognition. The experimental process is shown in Figure 5.

4.1. Experiment Results of Gesture Extraction. According to the design process of Section *2*, the improved threshold segmentation method is used for gesture extraction, and the flow chart is shown in Figure 6.

According to the flow chart, the real-time recognition results of the gesture extraction at different distances are shown in Figure 7. From Figure 7, it can be seen that the improved threshold segmentation method can automatically and efficiently identify the position of hand gesture in real time with 0.6 m, 1.0 m, 1.5 m, and 2.0 m, respectively, and the recognition distance is relatively wider than that in the existing literature [4, 5, 25]. Furthermore, clear hand gestures in ROI region can be obtained by using the improved method, which lead to a higher accuracy of the fingertip detection.

4.2. Experiment Results of Fingertip Detection and Gesture Recognition. According to the presented method in the Section *3*, the fingertip positions of the human hand can be detected, and the marked pixel coordinates are shown in Figure 8.

Figure 8(a) shows the result of the fingertip detection, where the red dot indicates the positions of the five fingertips. Figure 8(b) shows the results of the positioning, where d1 to d5 are the pixel distance values between five fingertips and the palm point, coordinate1 to coordinate5 are the coordinates of five fingertips in the pixel coordinates of image *P*,
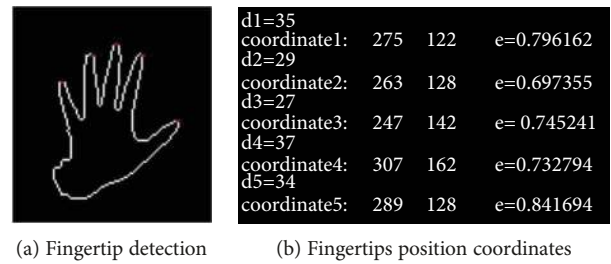

(a) Fingertip detection   (b) Fingertips position coordinates

Figure 8: Fingertip detection and positioning results.

Table 1: Experiment results in fingertip detection.

Distance	e	Total	Correct	Correct rate (%)
0.5 m–0.8 m	$0.65 < e < 1$	105	104	99.05
0.8 m–1.2 m	$0.65 < e < 1$	105	105	100.00
1.2 m–1.6 m	$0.65 < e < 1$	105	103	98.10
1.6 m–2.0 m	$0.65 < e < 1$	105	102	97.14

and e is the k-cosine curvature corresponding to the previous fingertip coordinate point.

We obtain the images in different distances and complex backgrounds from Kinect sensor in real time and extract the images of five fingers open gesture using the improved threshold segmentation method. The results of the fingertip detection are shown in Table 1, where "correct" means the five fingertips can be found and positioned correctly.

From Table 1, it can be seen that hand gesture is identified and fingertips are positioned between 0.5 m and 2.0 m in a complex background. Actually, the detection rate is relatively high. At a distance of about 1.0 m, the best positioning performance of fingertip detection is achieved. However, the recognition speed is reduced when the distance is too close or too far. Some small noise spots around the hands will reduce the detection performance of fingertips when the distance is
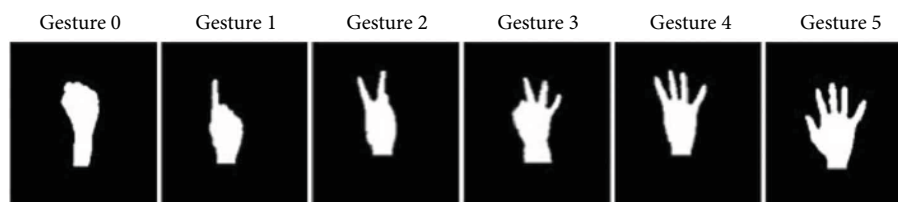
FIGURE 9: Six kinds of hand gestures.

too close, and the contour sequence points in the ROI are so less that there are not enough data to detect the five fingertips when the distance is too far.

According to the fingertip positioning, 650 images including six gestures from real-time videos were randomly selected to recognize hand gesture in the condition of different depth distances and different backgrounds from five experimental participants. The six kinds of gestures which represent numbers 0~5 are shown in Figure 9. Then, the identification results of gesture numbers 0~5 are shown in Table 2, where "recognition rate" indicates that the test results are the same as those of the extended gesture.

According to the experiment results of fingertip detection and gesture recognition, hand gesture identification can be achieved by the improved threshold segmentation method between 0.5 m and 2.0 m in real time, which is a relatively longer distance gesture identification in comparison to the direct distance interval setting. For example, the distance interval is set between 0.8 m and 1.0 m in [25]. Moreover, the proposed method has good recognition performance in a complex forefront and background, and the recognition rate of the number gesture experiment in this paper further shows that the improved method can not only identify the hand in real time in a complex background and at different distance but also meet the requirement of fingertip detection and gesture recognition to achieve a natural human-robot interaction.

5. Conclusions

Aiming at the distance limitation of gesture recognition, this paper proposes an improved threshold segmentation method with depth information for hand gesture segmentation and presents the k-cosine curvature algorithm for fingertip detection. First, the improved threshold segmentation method, which is a spatial stratification scanning method combined depth information with skin color RGB interval, is used to identify the position of hand gestures in a long distance. Second, the k-cosine curvature algorithm is used to detect the convex hull of fingers so as to determine the positioning of fingertips, and the numbers 0~5 of hand gesture can be thus identified. Third, the experimental results show that the proposed method can efficiently increase the detection distance in comparison to the traditional threshold segmentation methods. Moreover, every fingertip can be basically detected in the ROI by the improved method, and the recognition rates are more than 96%. Finally, the experiment results of number gesture recognition also show that the proposed method can meet the requirement of hand gesture recognition at different distances. Further work will be devoted to

TABLE 2: Number gesture recognition.

Gesture	0	1	2	3	4	5
Recognition rate (%)	96.67	96.72	98.88	96.15	98.15	97.67

identify more gesture information, to apply to the human-machine interaction, and to achieve more machine control function by dynamic and static gesture recognition.

Abbreviations

ROI: Region of interest
2D: 2 dimensional
3D: 3 dimensional
RGB: Red green blue
IR: Infrared radiation.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

References

[1] R. Huang and G. Shi, "Design of the control system for hybrid driving two-arm robot based on voice recognition," in *IEEE 10th International Conference on Industrial Informatics*, pp. 602–605, Beijing, China, July 2012.

[2] Y. Liu, Y. Li, X. Ma, and R. Song, "Facial expression recognition with fusion features extracted from salient facial areas," *Sensors*, vol. 17, no. 4, p. 172, 2017.

[3] W. Takano and Y. Nakamura, "Action database for categorizing and inferring human poses from video sequences," *Robotics and Autonomous Systems*, vol. 70, pp. 116–125, 2015.

[4] D. Q. Leite, J. C. Duarte, L. P. Neves, J. C. de Oliveira, and G. A. Giraldi, "Hand gesture recognition from depth and infrared Kinect data for CAVE applications interaction,"

Multimedia Tools and Applications, vol. 76, no. 20, pp. 20423–20455, 2017.

[5] X. L. Guo and T. T. Yang, "Gesture recognition based on HMM-FNN model using a Kinect," *Journal on Multimodal User Interfaces*, vol. 11, no. 1, pp. 1–7, 2017.

[6] V. Gonzalez-Pacheco, M. Malfaz, F. Fernandez, and M. Salichs, "Teaching human poses interactively to a social robot," *Sensors*, vol. 13, no. 12, pp. 12406–12430, 2013.

[7] D. Sidobre, X. Broquère, J. Mainprice et al., "Human–robot interaction," in *Advanced Bimanual Manipulation*, B. Siciliano, Ed., vol. 80 of Springer Tracts in Advanced Robotics, pp. 123–172, Springer, Berlin, Heidelberg, 2012.

[8] G. Kollegger, M. Ewerton, J. Wiemeyer, and J. Peters, "BIMROB – bidirectional interaction between human and robot for the learning of movements," in *Proceedings of the 11th International Symposium on Computer Science in Sport (IACSS 2017)* of Advances in Intelligent Systems and Computing, pp. 151–163, Springer, Cham, September 2018.

[9] M. Daushan, R. Thenius, K. Crailsheim, and T. Schmickl, "Organising bodyformation of modular autonomous robots using virtual embryogenesis," in *New Trends in Medical and Service Robots. MESROB 2016*, Mechanisms and Machine Science, vol 48, M. Husty and M. Hofbaur, Eds., pp. 73–86, Springer, Cham, 2018.

[10] T. Petrič, M. Cevzar, and J. Babič, "Shared control for human-robot cooperative manipulation tasks," in *Advances in Service and Industrial Robotics. RAAD 2017*, Mechanisms and Machine Science, vol 49, C. Ferraresi and G. Quaglia, Eds., pp. 787–796, Springer, Cham, 2018.

[11] M. Panwar, "Hand gesture recognition based on shape parameters," in *2012 International Conference on Computing, Communication and Applications*, pp. 1–6, Dindigul, Tamilnadu, India, Febuary 2012.

[12] B. K. Chakraborty, M. K. Bhuyan, and S. Kumar, "Fusion-based skin detection using image distribution model," in *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing - ICVGIP '16*, p. a67, Guwahati, Assam, India, December 2016.

[13] W. Song, D. Wu, Y. Xi, Y. W. Park, and K. Cho, "Motion-based skin region of interest detection with a real-time connected component labeling algorithm," *Multimedia Tools & Applications*, vol. 76, no. 9, pp. 11199–11214, 2016.

[14] R. L. Vieriu, I. Mironica, and B. T. Goras, "Background invariant static hand gesture recognition based on hidden Markov models," in *2013 International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, Iasi, Romania, July 2013.

[15] Y. W. Lee, "Implementation of an interactive interview system using hand gesture recognition," *Neurocomputing*, vol. 116, no. 10, pp. 272–279, 2013.

[16] A. Królak, "Use of Haar-like features in vision-based human-computer interaction systems," in *2012 Joint Conference New Trends in Audio & Video and Signal Processing: Algorithms, Architectures, Arrangements, and Applications (NTAV/SPA)*, pp. 139–142, Lodz, Poland, September 2015.

[17] J. Yang, R. Xu, Z. Ding, and H. Lv, "3D character recognition using binocular camera for medical assist," *Neurocomputing*, vol. 220, no. 11, pp. 17–22, 2017.

[18] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[19] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: structured-light versus time-of-flight Kinect," *Computer Vision and Image Understanding*, vol. 139, pp. 1–20, 2015.

[20] J. Jancek, D. Aleinikava, and G. M. Mirsky, "Optimizing Kinect® depth sensing using dynamic polarization," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE '17*, pp. 767-768, Seattle, WA, USA, March 2017.

[21] A. H. Smeragliuolo, N. J. Hill, L. Disla, and D. Putrino, "Validation of the leap motion controller using markered motion capture technology," *Journal of Biomechanics*, vol. 49, no. 9, pp. 1742–1750, 2016.

[22] C. Tang, G. Y. Tian, X. Chen, J. Wu, K. Li, and H. Meng, "Infrared and visible images registration with adaptable local-global feature integration for rail inspection," *Infrared Physics & Technology*, vol. 87, pp. 31–39, 2017.

[23] O. Ťupa, A. Procházka, O. Vyšata et al., "Motion tracking and gait feature estimation for recognising Parkinson's disease using MS Kinect," *Biomedical Engineering Online*, vol. 14, no. 1, p. 97, 2015.

[24] A. Procházka, O. Vyšata, M. Vališ, O. Ťupa, M. Schätz, and V. Mařík, "Bayesian classification and analysis of gait disorders using image and depth sensors of Microsoft Kinect," *Digital Signal Processing*, vol. 12, no. 47, pp. 169–177, 2015.

[25] C. Youness and M. Abdelhak, "Machine learning for real time poses classification using Kinect skeleton data," in *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pp. 307–311, Beni Mellal, Morocco, March-April 2016.

[26] X. Lu, C. C. Chen, and J. K. Aggarwal, "Human detection using depth information by Kinect," in *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 15–22, Colorado Springs, CO, USA, June 2011.

[27] C. M. Oh, M. Z. Islam, J. S. Lee, C. W. Lee, and I. S. Kweon, "Upper body gesture recognition for human-robot interaction," in *Human-Computer Interaction, Interaction Techniques and Environments. HCI 2011*, Lecture Notes in Computer Science, pp. 294–303, Springer, Berlin, Heidelberg, 2011.

[28] T. Wei, Y. Qiao, and B. Lee, "Kinect skeleton coordinate calibration for remote physical training," in *MMEDIA 2014 : The Sixth International Conferences on Advances in Multimedia*, pp. 66–71, Nice, France, 2014.

[29] M. Hamda and A. Mahmoudi, "Hand gesture recognition using Kinect's geometric and HOG features," in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications - BDCA'17*, pp. 1–5, Tetouan, Morocco, March 2017.

[30] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM international conference on Multimedia - MM '11*, pp. 1093–1096, Scottsdale, Arizona, USA, November–December 2011.

[31] Y. Zhang, C. Wang, J. Zhao, L. Zhang, and S.-C. Chan, "Template selection based superpixel earth mover's distance algorithm for hand gesture recognition," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 1002–1005, Chengdu, China, November 2017.

[32] A. Ramey, V. Gonzalez-Pacheco, and M. A. Salichs, "Integration of a low-cost rgb-d sensor in a social robot for gesture recognition," in *Proceedings of the 6th international conference on Human-robot interaction - HRI '11*, pp. 229-230, Lausanne, Switzerland, March 2011.

[33] R. M. Gurav and P. K. Kadbe, "Real time finger tracking and contour detection for gesture recognition using OpenCV," in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 974–977, Pune, India, May 2015.

[34] A. Kaehler, *Learning OpenCV*, O'Reilly Media, Sebastopol, CA, USA, 2008.