Vision-based Hand Recognition Based on ToF Depth Camera

Chao-Wei Yu, Chin-Hsuan Liu, Yen-Lin Chen, Posen Lee, and Meng-Syue Tian

Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan; Department of Occupational Therapy, I-Shou University, Kaohsiung, Taiwan

ABSTRACT

The current gesture recognition methods mostly adopt the classification-based approaches such as Neural Network (NN), Support Vector Machine (SVM), Hidden Markov Model (HMM) etc. As for the input image features, most research studies combined the color and depth images (ex. RGB-D) to obtain more accurate information of hand area, and such techniques may cost high computational resources and energy consumptions. To provide a low-cost gesture recognition method for wearable devices, this thesis used merely the Time-of-Flight depth camera to achieve a lightweight gesture recognition method. In most traditional gesture recognition methods, users have to wear gloves or bracelets to let depth cameras being able to accurately capture hands areas, and so that the hand contours, palm's distances, and angle feature can be obtained. Moreover, the Earth Mover’s Distance (EMD) algorithm, which is adopted in most gesture recognition approaches, costs high computational times. In this study, to avoid wearing gloves or bracelets, we propose a new algorithm that can compute the wrist cutting edges and capture the palm areas. In addition, this thesis proposes an efficient finger detection algorithm to judge the number of fingers, and significantly reduce the computing times. In the experimental results, our proposed method achieves a recognition rate of 90% and the performance has 5 frames per second on NVIDIA TX1 embedded platforms.

1. Introduction

Human–computer interaction (HCI) is a popular topic, which means the machine operation modes are gradually being developed and paid attention. Along with machine intelligence widely used, many devices are integrated with computing platform to promote users interest and interactive opportunity with machines, so HCI is an important issue. People want to use hand to control something and reduce dependence on other devices; therefore, gesture recognition becomes an important direction of research. The following will illustrate gesture recognition used in major areas in recent years. The following will illustrate gesture recognition application in major areas in recent years.

In 2016, Consumer Electronics, the vehicle manufacturers of BMW, showed the concept cars that can be controlled by gestures [1]. Based on the concept, users can use gestures to control the entertainment device or air-conditioning device when vehicles are in automatic mode, that will improve comfort and convenience when users control the vehicle peripherals systems.

In the field of virtual reality, users have to wear head-mounted display to play at virtual reality. So in such situations, medium through which to interact with virtual reality will be an important issue. The company ‘eyeSight’ is specialized in machine vision and gesture recognition, which provide convenience interface on many digital
2. Materials and Methods

On gesture recognition study based on image process, some methods combine RGB features such as skin color detection [14] to remove the area except hand area. [5] uses OpenNI SDK to capture and track hand area. Compared to these methods, the study only relies on depth image to achieve gesture recognition.

2.1. System Architecture

The study system flow is shown in Figure 1. The system captured the 165 × 120 pixels depth image and used normalization, binarization, median filter, and morphological to calculate the palm center and then through static and dynamic gesture recognition method to judge gestures.

2.2. Palm Center Detection

Here, we will use morphological to capture the palm center; first, through Opening operation to erode hand area, shown as Figure 2(b), then use dilation to dilate the block, shown as Figure 2(c), that is similar to the palm block; As a result, we can calculate approximate ellipses for the block and use the ellipse center be the palm center, as the blue point is palm center shown in Figure 2(d).

2.3. Finger Detection

We use Suzuki [15] to calculate the large block contour of the binarization image, as shown in Figure 3(b). When the block size is greater than a certain threshold value, it will judge the block as a hand area possibly; and then we use Ramer–Douglas–Peucker algorithm [16] to reduce the number of contour points, as shown in Figure 3(c).

2.4. Hand Segmentation

Shown as in Figure 4, when at different distance from depth camera, the arm length will be different and have a change of angle feature to affect the EMD calculation results and reduce the recognition success rate. Therefore, the study implemented the Hand segmentation method to
remove the arm and keep the palm. So that the difference in the feature of different distances will be narrow. When we observed the hand contour can be found which has to find out the two reference points WPa and WPb that can cut off the arm area, we set a sliding window to scan from the image bottom, as shown in Figure 5(a). The purpose is to check the width difference of white pixels on a certain interval. Suppose SW is sliding window, as shown in Figure 5(b) that will scan from the bottom to point P and judge the hand contour gradient change. When it has largest change, which is the reference point, we can succeed in removing the arm, as shown in Figure 5(c).

2.5. Feature Extraction and EMD Calculation

After Hand segmentation, we can calculate hand area feature histogram to be the EMD feature set. First, this study will focus on the input image and database image of contour points and palm center, respectively, to calculate the Euclidean distance and relative angle to be image feature set. Suppose the contour set is \( \{CP_1, CP_2, \ldots, CP_i\} \), the \( i \) is contour points number, as shown in Figure 6(a). The Euclidean distance of contour point
$\text{CPi}$ and palm center is defined as $E_{di}$. The Euclidean distance function is (1). After calculating the Euclidean distance, it also has to be normalized between 0 and 255, and recorded the maximum and minimum distances of the contour points, at last, normalizing distances of the contour points (2).

After calculating the Euclidean distance, we will calculate the angle of contour points with palm center. Figure 6(b) shows the palm center is center point and point O is starting point, then calculated the angle $\theta_i$ of contour point $\text{CPi}$ with palm center $P$ (3).

$$E_{di} = \sqrt{(CP_i \cdot x - P \cdot x)^2 + (CP_i \cdot y - P \cdot y)^2} \quad (1)$$

$$E_{di} = \frac{E_{di} - \text{minimumDist}}{\text{maximumDist} - \text{minimumDist}} \times 255 \quad (2)$$

$$\theta_i = \frac{\text{atan2}(CP_i \cdot y - P \cdot y, CP_i \cdot x - P \cdot x) \times 180}{\pi} + 180 \quad (3)$$

Table 1. Personal computer specification.

| Type       | Item                                      | Specification                     |
|------------|-------------------------------------------|-----------------------------------|
| Hardware   | CPU                                       | Intel(R) Core(TM) i7-3770 CPU 3.40 GHz |
|            | RAM                                       | DDR3 12G                          |
|            | Depth Camera                              | PMD Nano                          |
| Software   | Operating system                          | Microsoft Windows 10              |
|            | Development tool                          | Microsoft Visual Studio 2013      |
|            | Library                                   | OpenCV 3.0, PMD Nano API          |

Table 2. PMD Nano specifications.

| Parameter            | Value                                      |
|----------------------|--------------------------------------------|
| Type of sensor       | PMD PhotonICS+ 19 k-S3                      |
| Framerate            | $\leq$90 fps                               |
| Field of view        | $90^\circ \times 68^\circ$                 |
| Dimensions           | $37 \times 30 \times 25$ mm                |
| Illumination wavelength | 850 nm                                     |
| Power supply         | USB Powered (5 V @ 500 mA)                 |

Figure 5. Hand segmentation.

Figure 6. Feature extraction.

Figure 7. PMD Nano depth camera.
3. Results

The study used the personal computer and a depth camera to do experiments, and defined ten gestures to experimental test.

3.1. Experimental Environment

The experimental environment specifications are shown in Tables 1 and 2.
The study defined gesture name, real image and depth image shown as Figures 8 and 9. In order to compare the differences between different EMD thresholds and distances of the experimental results, we used different EMD thresholds and distances to experiment. The experiment will set three EMD thresholds as 10, 14, and 18, and distances as 15, 20, 25 cm from camera (Tables 3–5).

After the experiment of different EMD thresholds, we can learn from this. When we increase the EMD thresholds that can upgrade the recognition rate, but the EMD thresholds increases above a certain value, it will not have a great effect on the recognition rate.

### Table 3. Different thresholds and distances 15 cm experiment.

| EMD threshold | Gesture 1 | Gesture 2 | Gesture 3 | Gesture 4 | Gesture 5 | Gesture 6 | Gesture 7 | Gesture 8 | Gesture 9 | Gesture 0 | Total |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|
| 10 (%)        | 100      | 100      | 100      | 98.41    | 100      | 100      | 100      | 100      | 100      | 100      | 92.96 |
| 14 (%)        | 100      | 100      | 100      | 95.45    | 100      | 100      | 100      | 100      | 100      | 100      | 98.89 |
| 18 (%)        | 100      | 100      | 100      | 96.96    | 100      | 100      | 100      | 100      | 100      | 100      | 99.03 |

### Table 4. Different thresholds and distances 20 cm experiment.

| EMD threshold | Gesture 1 | Gesture 2 | Gesture 3 | Gesture 4 | Gesture 5 | Gesture 6 | Gesture 7 | Gesture 8 | Gesture 9 | Gesture 0 | Total |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|
| 10 (%)        | 100      | 100      | 100      | 98.24    | 100      | 100      | 100      | 100      | 100      | 100      | 91.16 |
| 14 (%)        | 100      | 100      | 100      | 98.18    | 100      | 100      | 100      | 100      | 100      | 100      | 97.87 |
| 18 (%)        | 100      | 100      | 100      | 94.54    | 100      | 100      | 100      | 100      | 100      | 100      | 97.29 |

### Table 5. Different thresholds and distances 25 cm experiment.

| EMD threshold | Gesture 1 | Gesture 2 | Gesture 3 | Gesture 4 | Gesture 5 | Gesture 6 | Gesture 7 | Gesture 8 | Gesture 9 | Gesture 0 | Total |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|
| 10 (%)        | 94.54    | 96.61    | 90.14    | 96.36    | 91.54    | 100      | 100      | 96.36    | 91.54    | 100      | 91.66 |
| 14 (%)        | 96.61    | 98.30    | 91.54    | 96.36    | 91.54    | 100      | 100      | 96.36    | 91.54    | 100      | 96.66 |
| 18 (%)        | 90.14    | 91.54    | 96.36    | 91.54    | 96.36    | 100      | 100      | 96.36    | 91.54    | 100      | 96.66 |

3.2. Gesture Experiment

The study defined gesture name, real image and depth image shown as Figures 8 and 9.

In order to compare the differences between different EMD thresholds and distances of the experimental results, we used different EMD thresholds and distances to experiment. The experiment will set three EMD thresholds as 10, 14, 18, and distances as 15, 20, 25 cm from camera (Tables 3–5).

After the experiment of different EMD thresholds, we can learn from this. When we increase the EMD thresholds that can upgrade the recognition rate, but the EMD thresholds increases above a certain value,
the increased recognition success rate will slow down, as shown in Figure 10 (Y-axis is the percentage of recognition rate).

This is followed by comparing the same user (User1) and the EMD threshold of 18 but in different distances situation, the user has or has not used the hand segmentation algorithm to identify gesture. The Table 6 is experiment video on different distances and the Tables 7–13 are the comparison tables at different distances.

Summary of the above experiment: in the distance of 15–25 cm, the effect of gesture recognition will be the best, but too close or too far will result in the poor quality of the image making the recognition rate decline.

The Table 14 is the gesture recognition rate for multiple users and the Table 15 is our method to compare with the one in [4]. In this paper, the method is based on hand segmentation to capture the palm region than doing the EMD calculation, but the reference [4] used the Thresholding Decomposition of the fixed threshold method. So the average recognition success rate of our method is 84.5%, and that of the one in [4] is 69.81%, which shows that the method of this study has a higher recognition success rate.

Table 9. The comparison table at 20 cm.

|Gesture| Without Hand segmentation | With Hand segmentation |
|-------|---------------------------|------------------------|
|0      | 100% (50/50)              | 100% (50/50)           |
|1      | 98.14% (53/54)            | 100% (54/54)           |
|2      | 100% (55/55)              | 100% (55/55)           |
|3      | 100% (61/61)              | 100% (61/61)           |
|4      | 100% (57/57)              | 100% (57/57)           |
|5      | 100% (55/55)              | 100% (55/55)           |
|6      | 93.22% (55/59)            | 100% (59/59)           |
|7      | 97.36% (74/76)            | 93.42% (71/76)         |
|8      | 100% (100/100)            | 100% (100/100)         |
|9      | 99.01% (706/713)          | 99.29% (708/713)       |

Table 10. The comparison table at 25 cm.

|Gesture| Without Hand segmentation | With Hand segmentation |
|-------|---------------------------|------------------------|
|0      | 100% (43/43)              | 97.67% (42/43)         |
|1      | 0% (0/55)                 | 96.35% (53/55)         |
|2      | 100% (59/59)              | 100% (59/59)           |
|3      | 100% (71/71)              | 91.54% (65/71)         |
|4      | 100% (68/68)              | 100% (68/68)           |
|5      | 100% (60/60)              | 100% (60/60)           |
|6      | 77.41% (48/62)            | 100% (62/62)           |
|7      | 69.33% (52/75)            | 98.66% (74/75)         |
|8      | 100% (95/95)              | 100% (95/95)           |
|9      | 100% (164/164)            | 100% (164/164)         |
|Total  | 87.76% (660/752)          | 98.67 (742/752)        |

Table 11. The comparison table at 30 cm.

|Gesture| Without Hand segmentation | With Hand segmentation |
|-------|---------------------------|------------------------|
|0      | 100% (46/46)              | 100% (46/46)           |
|1      | 0% (0/58)                 | 53.44% (31/58)         |
|2      | 96.49% (55/57)            | 82.45% (47/57)         |
|3      | 100% (65/65)              | 69.23% (45/65)         |
|4      | 100% (70/70)              | 55.71% (39/70)         |
|5      | 97.26% (71/73)            | 35.61% (26/73)         |
|6      | 0% (0/69)                 | 0% (0/69)              |
|7      | 0% (0/74)                 | 0% (0/74)              |
|8      | 77.31% (75/97)            | 77.31% (75/97)         |
|9      | 0% (0/144)                | 0% (0/144)             |
|Total  | 20.11% (382/753)          | 41.03% (309/753)       |

Table 12. The comparison table at 35 cm.

|Gesture| Without Hand segmentation | With Hand segmentation |
|-------|---------------------------|------------------------|
|0      | 0% (0/44)                 | 0% (0/44)              |
|1      | 0% (0/52)                 | 0% (0/52)              |
|2      | 0% (0/55)                 | 0% (0/55)              |
|3      | 0% (0/71)                 | 0% (0/71)              |
|4      | 0% (0/65)                 | 0% (0/65)              |
|5      | 0% (0/63)                 | 0% (0/63)              |
|6      | 0% (0/61)                 | 0% (0/61)              |
|7      | 0% (0/110)                | 0% (0/110)             |
|8      | 89.13% (82/92)            | 89.13% (82/92)         |
|9      | 0% (0/150)                | 0% (0/150)             |
|Total  | 10.74% (82/763)           | 10.74% (82/763)        |

Table 13. The comparison table at 40 cm.

|Gesture| Without Hand segmentation | With Hand segmentation |
|-------|---------------------------|------------------------|
|0      | 0% (0/46)                 | 0% (0/46)              |
|1      | 0% (0/62)                 | 0% (0/62)              |
|2      | 0% (0/60)                 | 0% (0/60)              |
|3      | 0% (0/58)                 | 0% (0/58)              |
|4      | 0% (0/61)                 | 0% (0/61)              |
|5      | 0% (0/60)                 | 0% (0/60)              |
|6      | 0% (0/49)                 | 0% (0/49)              |
|7      | 0% (0/64)                 | 0% (0/64)              |
|8      | 0% (0/114)                | 0% (0/114)             |
|9      | 0% (0/135)                | 0% (0/135)             |
|Total  | 0% (0/709)                | 0% (0/709)             |

Table 14. Multi-user.

|User| Recognition rate (%)|
|----|----------------------|
|User2| 79.05                |
|User3| 72.09                |
|User4| 81.49                |
|User5| 74.28                |
|User6| 80.91                |
|User7| 68.72                |
|User8| 87.78                |

Table 15. Methods comparison.

|                      | Our proposed EMD + Hand segmentation | [4] FEMD + Thresholding decomposition |
|----------------------|--------------------------------------|--------------------------------------|
|Average recognition  | 84.5%                                | 69.81%                               |
4. Conclusions

Integrating finger detection [5], EMD and hand segmentation can achieve a recognition rate of 80% on various users. In this situation of too close or too far from the camera, the input image quality will be poor and the angle features of gesture will be more different. So after using the Hand segmentation method, it will improve the recognition rate of 30%. It will be seen from this that the Hand segmentation method can improve the gesture recognition success rate. Apart from the Hand segmentation method, the thesis used the depth camera in the wearable situation and defined the gesture class that applicable in the wearable situation. The thesis is a great contribution to the field of wearable systems.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Ministry of Science and Technology of Taiwan under the grant number [MOST-105-2221-E-027-079], and University System of Taipei Joint Research Program under the grant numbers [USTP-NTUT-TMU-104-06, and USTP-NTUT-TMU-105-07].

ORCID

Chin-Hsuan Liu http://orcid.org/0000-0001-6285-1802

References

[1] “CES 2016: BMW shows off gesture-controlled concept car” [cited 2016 Jun 8]. http://www.bbc.com/news/technology-35258680
[2] “eyesight | ABOUT US” [cited 2016 Jun 7]. http://www.eyesight-tech.com
[3] “EyeSight demos VR gesture control using standard phone hardware” [cited 2016 May 17]. https://techcrunch.com/2016/05/17/eyesight-vr/
[4] Ren Z, Yuan J, Zhang Z. Robust hand gesture recognition based on finger earth mover’s distance with a commodity depth camera. Proceedings of the 19th ACM International Conference on Multimedia, Scottsdale, AZ, USA; 2011. p. 1093–1096.
[5] Lai Y-C. A technique of real-time gesture recognition based on depth image and skin color. Taiwan: National Taipei University of Technology, Department of Computer Science and Information Engineering; 2013.
[6] Kim J-H. 3-D hand motion tracking and gesture recognition using a data glove. Seoul: IEEE International Symposium on Industrial Electronics; 2009. p. 1013–1018.
[7] Biswas KK. Gesture recognition using Microsoft Kinect®. 5th International Conference on Automation, Robotics and Applications (ICARA), Wellington, New Zealand; 2011. p. 100–103.
[8] Chen Z-H, Kim J-T, Liang J, et al. Real-time hand gesture recognition using finger segmentation. Sci World J. 2014. Article ID. 267872, 9 pages.
[9] Chen F-S, Fu C-M, Huang C-L. Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Vision Comp. 2003;21(8):745–758.
[10] Huang D-Y, Hu W-C, Chang S-H. Vision-based hand gesture recognition using PCA+gabor filters and SVM. Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP ‘09. Kyoto, Japan; 2009. p. 1–4.
[11] Murakami K, Taguchi H. Gesture recognition using recurrent neural networks. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA; 1991. p. 237–242.
[12] Hasan H, Abdul-Kareem S. Static hand gesture recognition using neural networks. Artificial Intelligence Review. 2014;41(1):147–181.
[13] Ren Z, Yuan J, Zhang Z. Robust hand gesture recognition based on finger earth mover’s distance with a commodity depth camera. Proceedings of the 19th ACM International conference on Multimedia, Scottsdale, AZ, USA; 2011. p. 1093–1096.
[14] Van den Bergh M. Combining RGB and ToF Cameras for Real-time 3D Hand Gesture Interaction. Kona, HI: IEEE Workshop on Applications of Computer Vision (WACV); 2011. p. 66–72.
[15] Suzukita S, Abe K. Topological structural analysis of digitized binary images by border following. Comput. Vis., Graph., Image Process. 1985;30(1):32–46.
[16] Prasad DK, Leung MKH, Quek C, Cho S-Y. A novel framework for making dominant point detection methods non-parametric. Image Vision Comp. 2012;30(11):843–859.