Multimed Tools Appl (2018) 77:209–223
DOI 10.1007/s11042-016-4265-6

# Hand gesture recognition based on concentric circular scan lines and weighted K-nearest neighbor algorithm

Yanqiu Liu[1] · Xiuhui Wang[1] · Ke Yan[1]

Received: 24 February 2016 / Revised: 14 November 2016 / Accepted: 13 December 2016 /
Published online: 20 December 2016

**Abstract** Human-computer interactions based on hand gestures are of the most popular natural interactive modes, which severely depends on real-time hand gesture recognition approaches. In this paper, a simple but effective hand feature extraction method is described, and the corresponding hand gesture recognition method is proposed. First, based on a simple tortoise model, we segment the human hand images by skin color features and tags on the wrist, and normalize them to create the training dataset. Second, feature vectors are computed by drawing concentric circular scan lines (CCSL) according to the center of the palm, and linear discriminant analysis (LDA) algorithm is used to deal with those vectors. Last, a weighted k-nearest neighbor (W-KNN) algorithm is presented to achieve real-time hand gesture classification and recognition. Besides the efficiency and effectiveness, we make sure that the whole gesture recognition system can be easily implemented and extended. Experimental results with a user-defined hand gesture dataset and multi-projector display system show the effectiveness and efficiency of the new approach.

**Keywords** Hand gesture recognition · Linear discriminant analysis · Tortoise model · Weighted K-nearest neighbor

✉ Ke Yan
yanke@cjlu.edu.cn

[1] College of Information Engineering, China Jiliang University, 258 Xueyuan Street, Hangzhou 310018, China

## 1 Introduction

Human hand gesture, as intuitively the most natural 'language' for human-beings, is probably the simplest but an important tool for human-computer interaction (HCI). Hand gestures are defined by a series of movement by hand or hand-arm combination, which are also referred as dynamic hand gestures [4]. Using computer vision techniques, hand gestures are recognized as input signals mapping to hand gesture classes in databases to control target objects or scenes through some computerized systems [18, 19]. With the fast spreading of computer vision technology and applications in both scientific and engineering fields, hand gesture recognition techniques are nowadays utilized in virtual reality, video conferences, robotics and many other cutting-edge research areas [14, 15, 17, 20].

Based on signal input techniques, hand gesture recognition methods can be categorized into two types: hardware-based (e.g. sensor equipments signal inputs) and hardware-independent (e.g. real-time surveillance camera video inputs). Hardware-based hand gesture recognition methods detect hand gesture differences by digital gloves with sensors [7, 22]. Those methods usually require testers to wear physical equipments with sensors, which in some sense means not user-friendly for HCI. Moreover, digital gloves are restricted by hardware problems such as size, shape, sensitivity and etc. All these problems restrict the spread of usage for digital gloves in real-world applications.

Hardware-independent hand gesture recognition methods detects human hand gestures by skin colors, hand outlines and wrist tags based on video inputs. The extracted hand gesture features are classified with recognized hand gesture classes in database to output the classification results. The hardware-independent hand gesture recognition methods can be further categorized into two types: model-based [13, 30] and non-model based [27] according to the feature extraction approaches in the pre-processing step. Compared to hardware-based hand gesture recognition methods, hardware-independent methods are more logical and acceptable in daily HCI. Recent works show high recognition accuracy and sustainable systems. Difficulties for hardware-independent hand gesture recognition methods include the complex transformation of human hands and instability of video information such as lighting and coloring of video frames.

In this study, we attack the difficulties in video-based hand gesture recognition, and propose a simple, easily implementable, efficient, effective and extendable hand gesture recognition system based on concentric circular scan lines (HGR-CCSL). The HGR-CCSL method can be divided into two steps. In the pre-processing step, by utilizing the skin color information and tags on wrists, we extract information from input videos and build training dataset including various hand gesture pictures. Based on a set of circles centered on the center of palm, we extract the features of training dataset. In the second step, which is the real-time hand gesture recognition process, we classify and recognize hand gestures by a trained weighted k-nearest-neighbor (W-KNN) classifier.

The proposing algorithm is based on our previous work [29], but is a more efficient and effective method for real-time hand gesture recognition. The main contributions of our work are listed below:

- **A more efficient and effective feature extraction method.** Based on the work in [29], we add a set of circles centered on the center of palm for extracting features of gestures.

The new feature extraction method is demonstrated to be more efficient and effective in the experiment section.

- **A more reliable gesture recognition method.** We adopt the weighted KNN method for the gesture recognition to replace the original genetic algorithm (GA) method in [29]. The new gesture recognition method achieves higher recognition rates, and therefore is more reliable for real-world applications.
- **An easily implementable and extendable algorithm.** The detailed system setup is simple enough for starter level readers to implement an efficient and effective HCI model, e.g. the simple five gestures user-defined database, the simple tortoise model and the primitive W-KNN algorithm. Each step of the proposed method can be extended to a more sophisticated approach for more complex applications.

## 2 Related works

Video-based hand gesture recognition methods, which extract hand gesture features from video frames with or without model and later classify the features with supervised or unsupervised learning technique, become incrementally popular in artificial intelligence, signal processing, computer vision and virtual reality fields [8, 16, 21]. In 2007, Wang [29] designed a simple tortoise model to recognize the basic human hand gestures. The tortoise model builds a feature space hand geometry and texture and is able to efficiently map the target hand gesture with recognized hand gesture in database. The disadvantage of tortoise model is that the recognition result is highly sensitive to lighting, therefore requires stable lightening environment. Yang et al. [35] proposed an static hand hand gesture recognition system based on hand gesture feature space. The method did not handle the case while the human face was overlapping with hands. The recognition accuracy rates dropped quickly while the hand gesture differences were small. Zhang et al. [38] proposed a mean shift dynamic deforming hand hand gesture tracking algorithm based on region growth. This method did not require modelling for hand gestures but was highly sensitive to preprocessing result. The recognition accuracy dropped quickly while the hand gestures change drastically. Yao et al. [36] introduced a framework of hand posture estimation based on RGB-D sensors. This method utilize the hand outlines to reduce the complexity of hand posture mapping and support real-time complex hand gesture recognition. However, this method was not able to handle hand gestures appearing with arm and body. The hand part cannot be properly cut and recognized. Morency et al. [23] built a latent-dynamic discriminative models (LDDM) to detect sequential hand gestures in a video file. Kurakin et al. [11] introduced a real-time dynamic gesture recognition technique using a depth sensor. The proposed method is efficient and robust to different gesture styles and orientations. Yin et al. [37] introduced a high-performance training-free approach for hand gesture recognition using Hidden Markov Model (HMM) and Dynamic Time Warping (DTW). Recently, Wu et al. [3, 31, 32] developed a dynamic gesture recognition system with the depth information. The features of hand are extracted and the static hand posture are classified using the support vector machine (SVM). Xie and Cao [33] presented an accelerometer-based user-independent hand gesture recognition method. A simple database of 24 gestures are used, including 8 basic gestures and 16 complex gestures. As a result, 25 features are extracted based on the kinematics characteristics of the gestures, and treated as input for the training process of a feed-forward neural network model. The testing gestures are recognized by passing through a similarity matching process. Pu et al. [24, 25] developed a communication-based wireless system to recognize gestures. The gesture information was extracted from communication-based wireless signals; and by using a proof-of-concept prototype, they demonstrated that the proposed system was capable to detect nine whole-body gestures under certain circumstances. Jadooki et al. [10] presented a fusion based neural network gesture recognition system. The extracted gesture features are enhanced by fusions before classified by an artificial neural network. Dinh et al. [5, 6] developed a hybrid hand gesture recognition method combining random forest (RF) with a rule-based system. Their newest experimental results show a average recognition rate of 97.80 % over ten hand number gestures from five different subjects.

## 3 Feature extraction of hand gestures

### 3.1 The tortoise model

We represent the hand using a model combining an ellipse and five rigid finger as shown in Fig. 1 for feature extraction of hand gestures. The model is named as tortoise model and is used to represent the basic features of human hands. The tortoise model is defined as follows:

$$y = f(r_1, r_2, n, L_1, ..., L_n, W_1, ..., W_n, \theta_1, ..., \theta_n, R, G, B);$$
$$\text{subject to :}$$
$$\begin{cases} r_1 \geq 1.5r_2 \\ n \in [0, 5] \\ 1.2r_1 \geq L_i \geq 0.3r_1, i = 1, ..., n \\ r_1 \geq 3.0W_i, i = 1, ..., n \\ \theta_i \in [0, 90], i = 1, ..., n \\ R \in [0, 255], G \in [0, 255], B \in [0, 255] \end{cases};$$

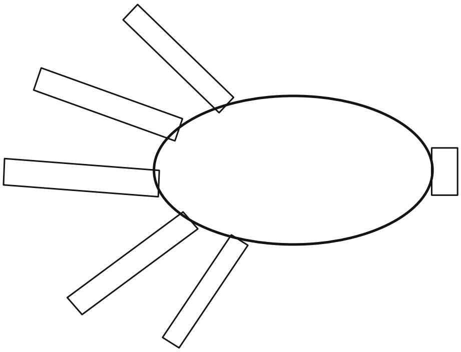


**Fig. 1** The tortoise model that represents a hand

where $r_1$, $r_2$ and n represent the radius of the palm, radius of the wrist and the number of fingers. $L_1, ..., L_n, W_1, ..., W_n$ represent the length and width of fingers. $\theta_1, ..., \theta_n$ represent the angle between the fingers and wrist. R, G and B represent the skin color.

The tortoise model has following three advantages:

1. **Ellipse representation for palm.** The ellipse representation for palm is simple and symmetric. It speeds up the hand shape segmentation and feature extraction process. The symmetry property of ellipse increases the false tolerance for hand rotations.
2. **Rectangle representations for fingers.** The rectangle representations for fingers provides the simplicity of calculating the finger postures by counting the number of fingers from different camera angles. The misjudgement of fingers because of overlapping can be effectively avoided.
3. **Relative length measurements for palm and finger length and width.** The relative length measurements ensure that the hands are always in right scale. And we use the relative measurements to distinguish different hand gestures. It prevents the hand scaling miscalculation while the arm is moving towards/away from the camera.

### 3.2 Hand gesture feature extraction based on circular scan lines

The hand gesture features can be extracted using concentric circles based on tortoise model (Fig. 2). The concentric circles are obtained by extracting the palm center $G$ and generating a set of circles centered at $G$ with radius $R_i$. The radius $R_i \in [R_{min}, R_{max}]$, where $R_{min}$ and $R_{max}$ denote the innermost circle radius and the outermost circle radius respectively. The value of $R_{min}$ is determined by an initial circle which intersects the outline of the hand. The value of $R_{max}$ is obtained by increasing $R_{min}$ by a small positive number $\delta$ until either of the two following conditions is reached:

1. The circle does not intersect the hand outline.
2. The circle does not intersect the image boundary.

Let the training image set be $T = \{t_{i,j} | i \in [1, C], j \in [1, N_i]\}$, where $C$ is the number of the hand gesture types and $N_i$ is the number of hand gesture samples in hand gesture type $i$. We obtained the optimized projection matrix $W$ using linear discriminant analysis (LDA)

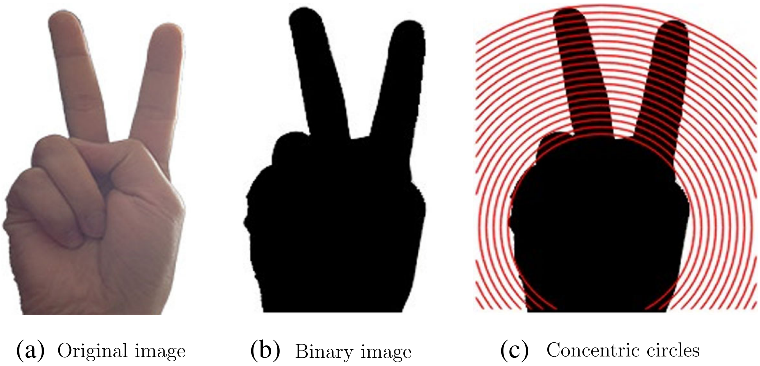

(a) Original image (b) Binary image (c) Concentric circles

**Fig. 2** Schematic diagram for extracting hand gesture features

[9, 12]. For every image sample $t_{i,j}$, $v_{i,j}$ and $v'_{i,j}$ are the feature vectors been processed before and after LDA. The detailed algorithm of hand gesture feature extraction based on circular scan lines is shown in Algorithm 1.

**Algorithm 1** Hand gesture feature extraction based on circular scan lines

**Input:** a series of training image, $T = \{t_{i,j}\}$.
**Output:** projection matrix $W$, dimension of feature vectors before projection transformation $D$ and feature vector set after projection transformation $V = \{v'_{i,j}\}$.

- **Step 1:** Segment the hand image (Fig. 2a). Based on the tortoise model, we segment the hand part image $T' = \{t'_{i,j} | i \in [1, C], j \in [1, N_i]\}$ from the input image using skin color information and the wrist tags. The skin color information extraction is done in the (Hue, Saturation and Value) HSV color space [2, 28]. The HSV color space is a well-known tool for separating hue, saturation and brightness. Moreover, the HSV color space is insensitive to lighting, therefore easy for skin color information collection.
- **Step 2:** Convert image $T'$ to binary image and scale it to a unified size (Fig. 2b). Due to the Euclidean distance between hand and camera, the scales of the extracted hand images are different. The normalization restore all hand images to pre-defined size, which is beneficial for later steps. We denote the normalized binary image as $T'' = \{t''_{i\ j} | i \in [1, C], j \in [1, N_i]\}$.
- **Step 3:** Extract the hand gesture features using concentric circles (Fig. 2c). Xie and Ji [34] proposed an algorithm to detect ellipses using Hough transformation. We use a similar approach to find the ellipse (palm) center from $T''$ and draw concentric circles to extract the hand gesture features. The extracted hand gesture features from the previous step are the number of concentric circles hitting the outline of hands, which is denoted as $P_x$.
- **Step 4:** Normalize all $P_x$ to dimension $D$. The extracted feature vectors may have different dimensions. In this step, we normalize the feature vectors to unified dimension $D$. Without loss of generality, we assume that the minimal dimension of all extracted feature vectors to be $n_1$. For feature vectors $P_x$ with dimension $n_2 > n_1$, we merge the inner most ($n_2$-$n_1 + 1$) dimensions using weighted addition. The weight decreases towards the center of palm. The normalized vector set is denoted as $V = \{v_{i,j}\}$, and $D = n_1$.
- **Step 5:** Output results. The optimized projection matrix $W$ is obtained by LDA, which projects the high dimensional data to the 'optimal' projection vector space. The definition of 'optimal' refers to the maximization of the inter-class distances and minimization of the intra-class distances. The final hand gesture feature vector set $V = \{v'_{i,j}\}$ is computed by transforming $V = \{v_{i,j}\}$ with $W$.

## 4 Hand gesture recognition based on W-KNN

The state-of-art KNN algorithm [1] is used to recognize hand gestures online. In this study, we utilize a small user-defined database including five gestures (Fig. 3). Therefore,

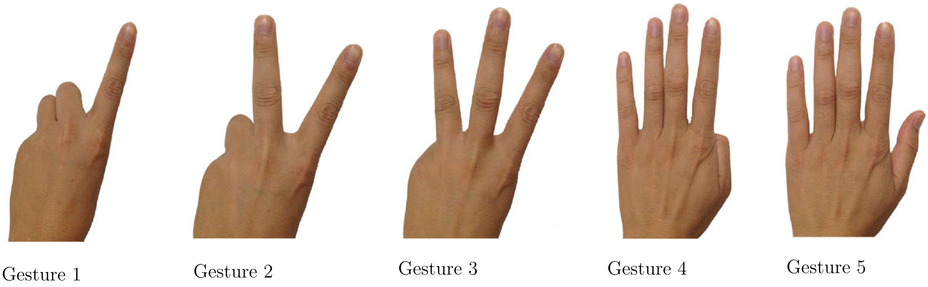


**Fig. 3** A small user-defined gesture database including five basic gestures: Gesture 1, 2, ..., 5

$K \in [1, 5]$, and for simplification purposes, we set $K = 3$ for all experiments in this study. Since the KNN is a fundamental machine learning method, the proposed algorithm (Algorithm 2) can be easily extended to other machine learning methods. For hand gesture recognition studies, we add an additional weight to each sample to denote the co-relation between samples within the same class. It is a common technique for relatively small training datasets; and can be easily extended to more sophisticated databases.

**Algorithm 2** Hand gesture recognition based on W-KNN

**Input:** testing image $I_x$, dimension of feature vectors before projection transformation $D$, projection matrix $W$ and feature vector set $V = \{v'_{i,j}\}$.
**Output:** the class of $I_x$: $C_x \in [1, C]$.

- **Step 1:** Extract the initial feature vector of the testing gesture. Repeat step 1 to 3 in Algorithm 1 to obtain the feature vector $P_x$ for $I_x$.
- **Step 2:** Normalize the dimension of $P_x$ to $D$. Suppose the original dimension of $P_x$ is $D_x$. If $D_x > D$, do the normalization similar to Step 2 in Algorithm 1; if $D_x < D$, increase the dimension of $P_x$ by adding $(D - D_x)$ 1s at the beginning of $P_x$ (the same effect as adding more circles inside the palm). The normalized feature vector is denoted as $V_x$.
- **Step 3:** Transform $V_x$ using the optimal projection matrix $W$. The transformed feature vector is denoted as $V'_x$.
- **Step 4:** Find K nearest neighbor of $V'_x$ by calculating the distance between each vector in $V'$ (result of Algorithm 1) and $V'_x$. The K nearest vector set is denoted as:

$$V_N = \{v'_{i,j,k} | i \in [1, C], j \in [1, N_i], k = 1, 2, ..., K\},$$

  where $C$ is the number of classes and $N_i$ is the number of samples for class $i$.
- **Step 5:** Compute the weight $W_p$ for each vector $v'_{i,j,p}$: $W_p = \frac{\phi_1}{\phi_2}$, where $\phi_1$ and $phi_2$ are the standard deviation for $V_N$, before and after inserting the testing feature vector $P_x$ into $V_N$.
- **Step 6:** Vote for classification. The acquired hand gesture class is obtained by weighted voting based on the k-nearest neighbor in $V_N$ for $P_x$.

## 5 Experiment and analysis

### 5.1 System setup

A multi-projector display system is built to verify the effectiveness of the HGR-CCSL method. The flowchart of the system is shown in Fig. 4. The most important component of the system are the pre-processing parts, namely hand tracing and hand image segmentation. We omit the details of the system control in this paper. Readers who have interest in the controlling part can refer to our previous work [29]. The system completes the hand gesture recognition in two steps: offline pre-processing and online hand gesture recognition.

The detail steps of offline pre-processing are:

1. **Data input**. The input dataset is a series of images showing meaningful hand gestures.
2. **Hand image segmentation**. The hand part is extracted as a separate image from input using skin color features and Gaussian mixture modelling for background subtraction [26, 39].
3. **Hand gesture extraction**. The training dataset is obtained by drawing a set of concentric circular circles intersecting the hand outline. The training dataset is processed by LDA and trained offline.

The offline pre-processing phase follows Algorithm 1. to obtain the training dataset for the classification phase. In the real-world applications, the offline pre-processing phase can be done in advance, and therefore not counted in efficiency measurement.

The detail steps of online hand gesture recognition are:

1. **Data input**. The testing data input images are from the real-time video cameras capturing user hand postures.
2. **Hand image segmentation**. Different from the offline pre-processing part, in this step, we have to consider various case, such as the scale and location of the hands. The co-relation of the video frames is considered in hand image extraction. The extracted images are translated and scaled to a unified size.
3. **Real-time hand gesture recognition**. The input hand gesture is recognized by trained model in offline pre-processing part using W-KNN (Algorithm 2).

In HGR-CCSL, the training model is trained offline, which increases the real-time hand gesture recognition speed. The W-KNN model improves the accuracy rate compared to other existing methods (Section 5.2).

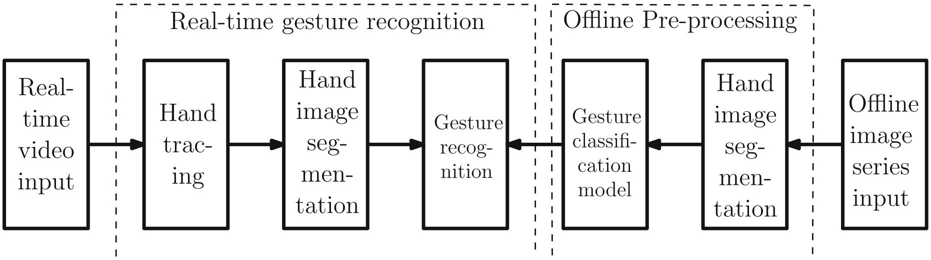

Real-time gesture recognition
Offline Pre-processing
Real-time video input
Hand trac-ing
Hand image seg-men-tation
Gesture recog-nition
Gesture classifi-cation model
Hand image seg-men-tation
Offline image series input

**Fig. 4** The hand gesture recognition system used in our experiment

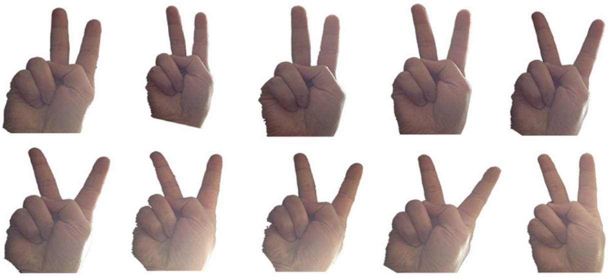


**Fig. 5** The ten variances of Gesture 2

## 5.2 Results

We compare the proposed HGR-CCSL algorithm with both hardware-based and hardware-independent hand gesture recognition approaches. The hardware-based approaches utilize a hand glove branded '5DT Data Glove Ultra' with five sensors, and include the digital glove systems developed in [22] and the accelerometer-based hand gesture recognition using neural network (AHGR-NN) [33]. The hardware-independent approaches include HGR-LDDM [23] and HGR-AGA [29]. We define a customized small gesture database to compare the effectiveness and efficiency of the above mentioned algorithms. The user-defined small database includes five gestures (Gesture 1 to 5 in Fig. 3); and each gesture includes ten variances for testing robustness (in Fig. 5, we show the ten variances of Gesture 2 for demonstration purposes). Figure 6 shows the experimenting environment, which involves a large screen with a multi-projector system. Three sets of hand gesture data, namely 'Selection', 'Translation' and 'Rotation', are defined by combinations of Gesture 1 to 5. The utilized digital camera is 'HISUNG IPC-EH5110PL-IR3'. The system environment is a 'HP Pro 3380 MT' personal computer with 4G RAM and an i5-3470 CPU.

The misclassification rates and average gesture recognition speed comparisons between the hardware-based approaches are shown in Table 1. We compare the misclassification rates in Fig. 7. The HGR-CCSL method has lower misclassification rates compared to methods using digital glove systems developed in [22], especially for hand gestures in 'Translation' and 'Rotation' datasets, where hand gesture differences are smaller. The misclassification rates of the AHGR-NN method is better than [22], but still higher than the proposed HGR-CCSL method. The reason of the high misclassification rates for hard-ware based methods is that the methods highly depend on the sensitivity of the hardware. It is

**Fig. 6** The testing environment involving a large projector system

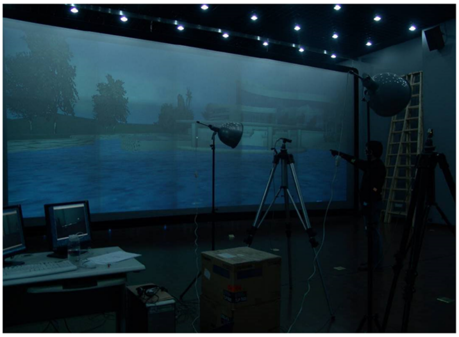


**Table 1** Misclassification rates and average gesture recognition speed comparison between HGR-CCSL, digital glove and AHGR-NN methods

| Method | Dataset | # testing samples | Misclass. rate (%) | Rec. speed (#ges./sec.) |
|---|---|---|---|---|
| HGR-CCSL | Selection | 200 | 0.9 | 10.1 |
| | Translation | 200 | 1.4 | 9.3 |
| | Rotation | 200 | 1.6 | 10.5 |
| Digital glove | Selection | 200 | 4.4 | 1.1 |
| | Translation | 200 | 9.2 | 2.7 |
| | Rotation | 200 | 9.7 | 3.0 |
| AHGR-NN | Selection | 200 | 1.2 | 5.8 |
| | Translation | 200 | 3.6 | 6.0 |
| | Rotation | 200 | 3.8 | 7.2 |

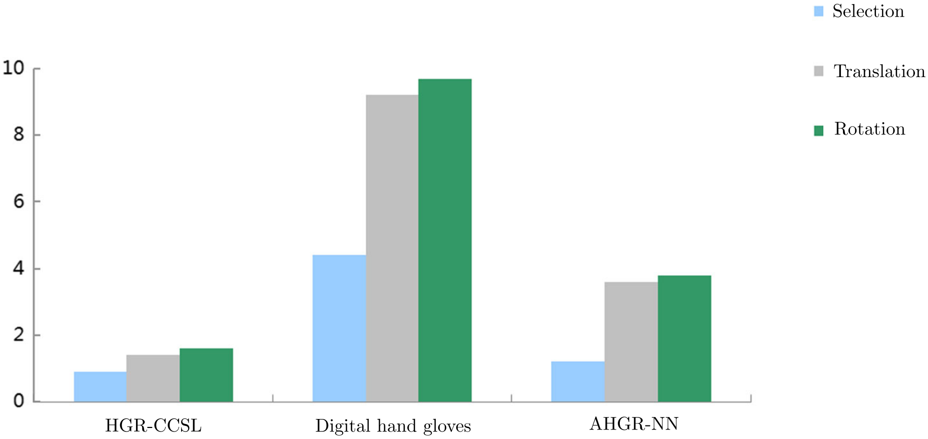


**Fig. 7** The misclassification rates comparison against hardware-based methods

**Table 2** Misclassification rates and average gesture recognition speed comparison between HGR-CCSL, HGR-LDDM and HGR-AGA

| Method | Dataset | # testing samples | Misclass. rate (%) | Rec. speed (#ges./sec.) |
|---|---|---|---|---|
| HGR-CCSL | Selection | 200 | 0.9 | 10.1 |
| | Translation | 200 | 1.4 | 9.3 |
| | Rotation | 200 | 1.6 | 10.5 |
| HGR-LDDM | Selection | 200 | 8.1 | 9.3 |
| | Translation | 200 | 11.3 | 8.9 |
| | Rotation | 200 | 12.0 | 9.1 |
| HGR-AGA | Selection | 200 | 3.1 | 16.3 |
| | Translation | 200 | 5.7 | 14.7 |
| | Rotation | 200 | 5.9 | 15.4 |

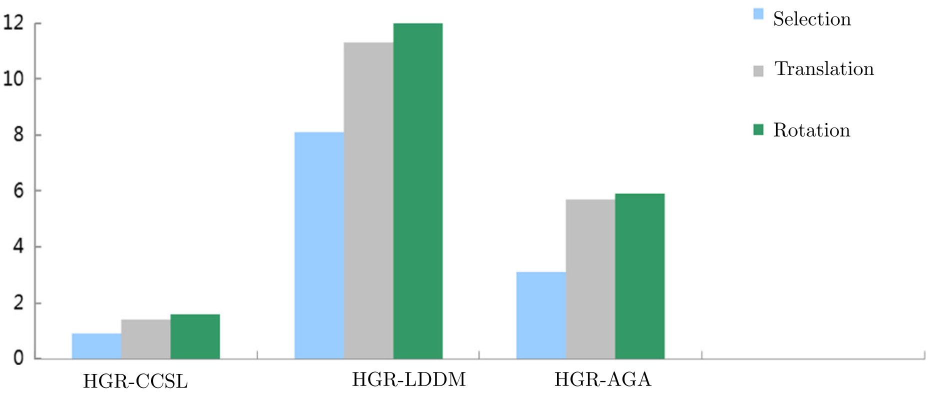


**Fig. 8** The misclassification rates comparison against hardware-independent methods

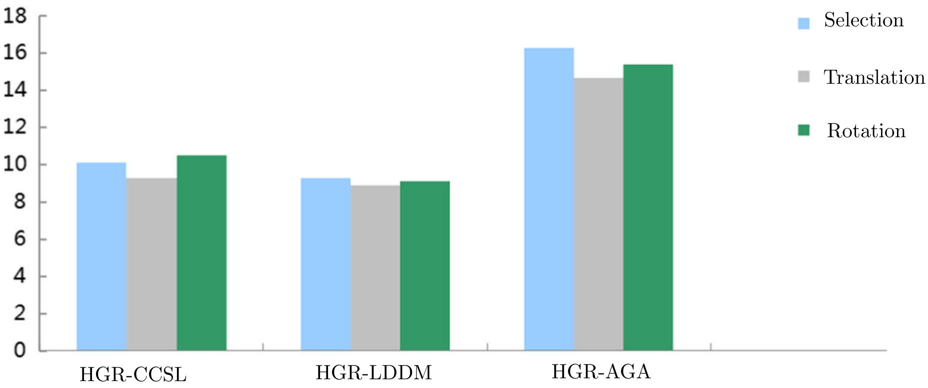
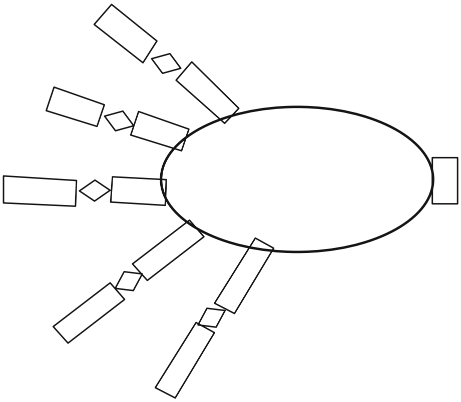


**Fig. 9** The average hand gesture recognition speed (No. of gestures/sec.) comparison against hardware-independent methods

**Fig. 10** The tortoise model in Fig. 1 can be extended to produce a larger database



also noted that the hardware-based methods are much more efficient methods speed-wise comparing with our method.

Second, we compare HGR-CCSL with HGR-LDDM and HGR-AGA. The results are shown in Table 2. We show the misclassification comparison in Fig. 8 and the efficiency comparison in Fig. 9. The HGR-CCSL method improves the hand gesture recognition misclassification rates based on HGR-AGA method but slows down the computation speed because of the W-KNN computation time. Compared to HGR-LDDM method, the HGR-CCSL method has lower recognition misclassification rates as well as faster computation speed.

## 6 Conclusion

In this study, we proposed a hand gesture feature extraction and recognition method based on concentric circular scan lines and weighted KNN algorithm. The proposed HGR-CCSL method can be divided into two parts, namely offline part and online part. Heavy computations such as hand part image segmentation, extraction of feature vectors from the concentric circular scan lines and training W-KNN classifier are processed in offline mode. The online hand gesture recognition part only deal with testing samples and is efficient enough for real-time applications. Experimental results show that the HGR-CCSL method has higher recognition accuracy rates with acceptable computational speed compared to existing methods.

The purpose of this study is to show a simple and easily re-implementable HCI approach for real-world applications. Every step of the proposed algorithm can be easily extended to more sophisticated systems according to customized requirements. For example, the simple tortoise model depicted in Fig. 1 can be easily extended to a more complex model by adding five joints to the fingers (Fig. 10). The small five gesture ‘user-defined’ database can be extended to involve more gestures. The W-KNN algorithm, which is known as a basic machine learning technology, can also be extended to other machine learning methods.

As a future work, we are enriching our gesture database by involving more gesture variations. The database, which we used in this work, may become publicly available in the near future.

**Acknowledgments** This work is supported by National Science Foundation of China (Numbers: 61303146, 61602431).

## References

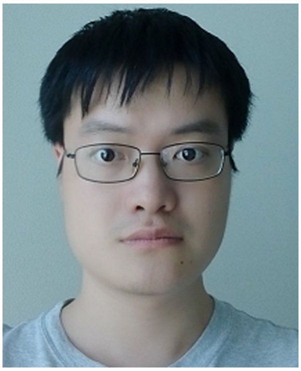
1. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185
2. Chen TW, Chen YL, Chien SY (2008) Fast image segmentation based on k-means clustering with histograms in hsv color space. In: IEEE 10th workshop on multimedia signal processing, 2008, IEEE, pp 322–325
3. Chen WL, Wu CH, Lin CH (2015) Depth-based hand gesture recognition using hand movements and defects. In: International symposium on next-generation electronics
4. Darrell TJ, Pentland AP (1994) Classifying hand gestures with a view-based distributed representation. In: Advances in neural information processing systems, pp 945–952
5. Dinh DL, Kim JT, Kim TS (2014) Hand gesture recognition and interface via a depth imaging sensor for smart home appliances. Energy Procedia 62:576–582
6. Dinh DL, Lee S, Kim TS (2016) Hand number gesture recognition using recognized hand parts in depth images. Multimedia Tools and Applications 75(2):1333–1348

7. Gregorio P (2008) Capacitive sensor gloves. US Patent App. 12/250,815
8. Hasan H, Abdul-Kareem S (2014) Human–computer interaction using vision-based hand gesture recognition systems: a survey. Neural Comput Applic 25(2):251–261
9. Huang R, Liu Q, Lu H, Ma S (2002) Solving the small sample size problem of lda. In: 16Th international conference on pattern recognition, 2002. Proceedings. IEEE, vol 3, pp 29–32
10. Jadooki S, Mohamad D, Saba T, Almazyad AS, Rehman A (2016) Fused features mining for depth-based hand gesture recognition to classify blind human communication. Neural Comput Applic:1–10
11. Kurakin A, Zhang Z, Liu Z (2012) A real time system for dynamic hand gesture recognition with a depth sensor, pp 1975–1979
12. Kyperountas M, Tefas A, Pitas I (2007) Weighted piecewise lda for solving the small sample size problem in face verification. IEEE Trans Neural Netw 18(2):506–519
13. Lee HK, Kim JH (1999) An hmm-based threshold model approach for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21(10):961–973
14. Lu Z, Lal Khan MS, Ur Réhman S (2013a) Hand and foot gesture interaction for handheld devices. In: Proceedings of the 21st ACM international conference on multimedia, ACM, pp 621–624
15. Lu Z et al. (2013b) Touch-less interaction smartphone on go! In: SIGGRAPH Asia 2013 Posters, ACM, p 28
16. Lv Z (2013) Wearable smartphone: wearable hybrid framework for hand and foot gesture interaction on smartphone. In: Proceedings of the IEEE international conference on computer vision workshops, pp 436–443
17. Lv Z, Li H (2015) Imagining in-air interaction for hemiplegia sufferer. In: International conference on virtual rehabilitation proceedings (ICVR), 2015, IEEE, pp 149–150
18. Lv Z, Réhman SU (2013) Multi-gesture based football game in smart phones
19. Lv Z, Halawani A, Lal Khan MS, Réhman SU, Li H (2013) Finger in air: touch-less interaction on smartphone. In: Proceedings of the 12th international conference on mobile and ubiquitous multimedia, ACM, p 16
20. Lv Z, Halawani A, Feng S, Li H, Réhman SU (2014) Multimodal hand and foot gesture interaction for handheld devices. ACM Trans Multimed Comput Commun Appl (TOMM) 11(1s):10
21. Lv Z, Halawani A, Feng S, Ur Réhman S, Li H (2015) Touch-less interactive augmented reality game on vision-based wearable device. Pers Ubiquit Comput 19(3-4):551–567
22. Mehdi SA, Khan YN (2002) Sign language recognition using sensor gloves. In: Proceedings of the 9th international conference on neural information processing, 2002. ICONIP'02. IEEE, vol 5, pp 2204–2206
23. Morency LP, Quattoni A, Darrell T (2007) Latent-dynamic discriminative models for continuous gesture recognition. In: IEEE Conference on computer vision and pattern recognition, 2007. CVPR'07. IEEE, pp 1–8
24. Pu Q, Gupta S, Gollakota S, Patel S (2013) Whole-home gesture recognition using wireless signals. In: Proceedings of the 19th annual international conference on mobile computing & networking, ACM, pp 27–38
25. Pu Q, Gupta S, Gollakota S, Patel S (2015) Gesture recognition using wireless signals. GetMobile: Mobile Computing and Communications 18(4):15–18
26. Reynolds DA, Quatieri TF, Dunn RB (2000) Speaker verification using adapted gaussian mixture models. Digital Signal Process 10(1):19–41
27. Suarez J, Murphy RR (2012) Hand gesture recognition with depth images: a review. In: RO-MAN, 2012 IEEE, IEEE, pp 411–417
28. Sural S, Qian G, Pramanik S (2002) Segmentation and histogram generation using the hsv color space for image retrieval. In: International conference on image processing. 2002. Proceedings. 2002, IEEE, vol 2, pp II–589
29. Wang X (2007) Gesture recognition based on adaptive genetic algorithm. Journal of Computer-Aided design and Computer Craphics 19(8):1056
30. Wilson AD, Bobick AF (1999) Parametric hidden markov models for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21(9):884–900
31. Wu CH, Lin CH (2013) Depth-based hand gesture recognition for home appliance control, pp 279–280
32. Wu CH, Chen WL, Lin CH (2015) Depth-based hand gesture recognition. Multimedia Tools and Applications:1–22
33. Xie R, Cao J (2016) Accelerometer-based hand gesture recognition by neural network and similarity matching. IEEE Sensors J 16(11):4537–4545
34. Xie Y, Ji Q (2002) A new efficient ellipse detection method. In: 16th international conference on pattern recognition, 2002. Proceedings. IEEE, vol 2, pp 957–960

35. Yang B, Song X, Feng Z, Hao X (2010) Gesture recognition in complex background based on distribution features of hand. Journal of Computer-Aided design and Computer Craphics 22(10):1841–1848
36. Yao Y, Fu Y (2012) Real-time hand pose estimation from RGB-D sensor. In: IEEE international conference on multimedia and expo (ICME), 2012, IEEE, pp 705–710
37. Yin L, Dong M, Duan Y, Deng W, Zhao K, Guo J (2014) A high-performance training-free approach for hand gesture recognition with accelerometer. Multimedia Tools and Applications 72(1):843–864
38. Zhang QY, Hu JQ, Zhang MY (2010) Mean shift dynamic deforming hand gesture tracking algorithm based on region growth. Pattern Recognition and Artificial Intelligence:4
39. Zivkovic Z (2004) Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004. IEEE, vol 2, pp 28–31

**Yanqiu Liu** obtained the bachelor’s degree from Zhengzhou University, Zhengzhou, China in 2000 and the master’s degree from Jiangnan University, Wuxi, China. She worked as a software engineer from 2000 to 2002. She is currently working as a lecturer in China Jiliang University. Her interested research area includes human-computer interaction, digital image processing and pattern recognition.

**Xiuhui Wang** was awarded a PhD and Masc (Research) in 2007 and 2003 from Zhejiang University. His research and teaching interests are focused on computer graphics, computer vision, and computer networks. He commenced working as an academic in the Department of Computer Science and Technology, China Jiliang University in 2007, firstly as a Lecturer then an associate professor in 2009.



**Ke Yan** completed both the Bachelor's and Ph.D. degree in National University of Singapore (NUS). He received his Ph.D. certificate in computer science in 2012 under the supervision of Dr. Ho-Lun Cheng. During the years between 2013 and 2014, he was a post-doctoral researcher in Masdar Institute of Science and Technology in Abu Dhabi, UAE. Currently, he serves as a lecturer in China Jiliang University, Hangzhou, China. His main research field includes computer graphics, computational geometry, data mining and machine learning.