Multimedia Tools and Applications (2019) 78:23469–23488
https://doi.org/10.1007/s11042-019-7660-y

Occlusion-robust bimanual gesture recognition by fusing multi-views

Geoffrey Poon[1] · Kin Chung Kwan[1] · Wai-Man Pang[1]

Received: 6 September 2018 / Revised: 27 January 2019 / Accepted: 15 April 2019 /
Published online: 4 May 2019

Abstract

Human hands are dexterous and always be an intuitive way to instruct or communicate with peers. In recent years, hand gesture is widely used as a novel way for human computer interaction as well. However, existing approaches target solely to recognize single-handed gesture, but not gestures with two hands in close proximity (bimanual gesture). Thus, this paper tries to tackle the problems in bimanual gestures recognition which are not well studied from the literature. To overcome the critical issue of hand-hand self-occlusion problem in bimanual gestures, multiple cameras from different view points are used. A tailored multi-camera system is constructed to acquire multi-views bimanual gesture data. By employing both shape and color features, classifiers are trained with our bimanual gestures dataset. A weighted sum fusion scheme is employed to ensemble results predicted from different classifiers. While, the weightings in the fusion are optimized according to how well the recognition performed on a particular view. Our experiments show that multiple-view results outperform single-view results. The proposed method is especially suitable to interactive multimedia applications, such as our two demo programs: a video game and a sign language learner.

Keywords Gesture recognition · Bimanual gesture · Multiview · Learning based recognition · Occlusion · Ensemble of classifiers

1 Introduction

Hand gesture is a conventional media for communication in human history apart from speech and voice, and is also considered as one of the most convenient and intuitive forms

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11042-019-7660-y) contains supplementary material, which is available to authorized users.

✉ Wai-Man Pang
wmpang@ieee.org

[1] School of Computing and Information Sciences, Caritas Institute of Higher Education, Tseung Kwan O, Hong Kong

of control and command in many multimedia applications. A proper introduction of hand gesture inputs to a multimedia application can certainly improves excitement and user engagement, such as gestures for invoking certain special actions in games or hand sign language for learning purpose. Among the existing hand gesture recognition approaches, vision-based methods are the most popular and suitable for wider range of applications as they require no tailored equipment or wearing of gloves.

The community continuously put most of the efforts in recognizing single-handed gesture such as Fig. 1a, and achieved rather satisfactory results. The recent works from Wu and Kang [40] as well as Ren et al. [29] are good examples which achieved robust recognition even in complex environment. However, there are fewer works target for gestures formed by two hands (*bimanual gesture*) such as Fig. 1b. Conventional commercial solutions also do not provide satisfactory recognition results. Figure 2 demonstrates a case of failure using the leap motion hand tracking solution. A bimanual "gun" gesture is performed, while a degenerated result is recognized. Potential application of bimanual gesture can be unlimited, while it is with high demand in games and educational multimedia applications for sign language learning.

This paper attempts to tackle the vision-based recognition of static bimanual gestures appearing in multimedia applications and games. Although methods for single-hand recognition are applicable to bimanual gesture, a serious problem that need to be solved is the hand-hand self-occlusion. It is similar to the challenges in two persons' interaction recognition work by Li and Leung [19] in which serious occlusion causing ambiguity. Inspired by [19], we follow the multiview approach in order to minimize the ambiguity introduced by self-occlusion, a multi-camera setup is constructed to cover a wider range of viewing angles. However, unlike [19], our multiview setup tries not to rely on RGB-D cameras; as classical RGB cameras is less bulky and widely available, so it is more ideal for a practical multiview system. By arranging the monocular RGB cameras in diversified viewing points, more features can be provided to compensate other views in the recognition.

Our method tries to extract the color and shape features in the hand gesture frames acquired from the multiple synchronized views. However, conflict may occur between the recognized gestures from different views. As a result, a fusion scheme of results from different views is designed. The scheme considered all the performance of a particular view to a certain target gesture, because some hand gestures may look similar from a certain view causing ambiguity in recognition. Then, our fusion scheme tries to introduce weightings

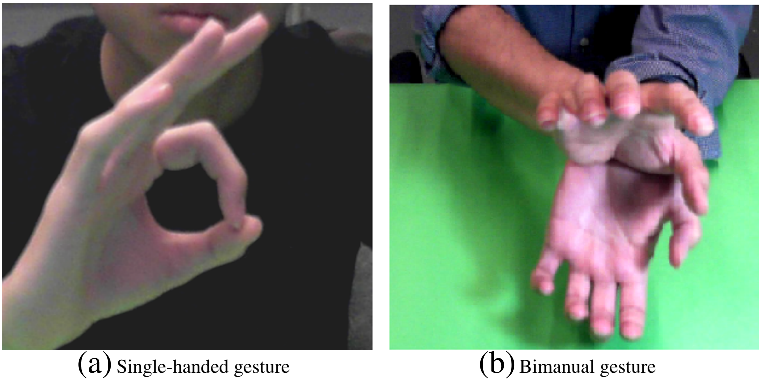

(a) Single-handed gesture (b) Bimanual gesture

Fig. 1 Examples of single-handed and bimanual gesture

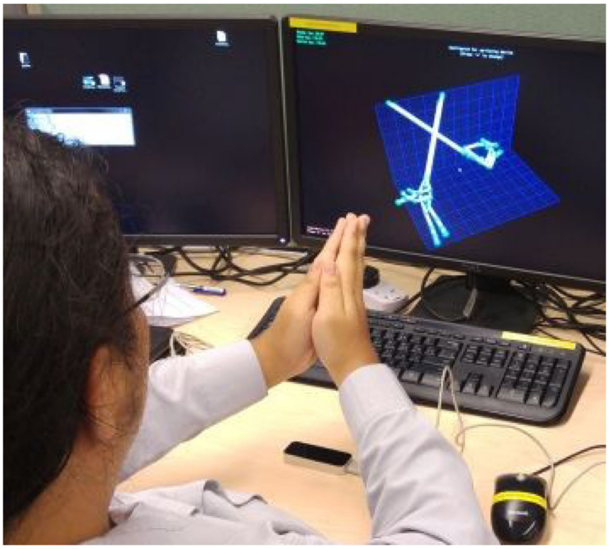


Fig. 2 Conventional commercial solution does not handle bimanual gesture well especially when significant occlusion occurs

so as to balance the contributions of different cameras for different gestures based on their performances. These weightings are obtained via a gradient decent based optimization process.

A total of 6 different bimanual gestures are used in our preliminary study. The experimental results reveal that, with the use of multiple views and an optimized weighting-based fusion scheme, the recognition accuracy is improved comparing to their single view counterparts. The proposed bimanual recognition is also deployed on a simple game and a sign language learning system to illustrate its interactive-rate capability.

In summary, this work tried to solve bimanual gesture recognition with a multi-view approach. With the construction of new multi-camera system and new dataset acquired, multiple classifers are ensembled to obtain better results. Furthermore, we proposed using optimization in seeking the best fusion scheme in our multiple view approach. All above mentioned will contribute to further research in bimanual gesture recogition.

The rest of this paper is organized as follows. Significant works from the literature of hand gesture recognition are reviewed in Section 2. In Section 3, we show our multi-view capturing setup for acquiring bimanual gesture data. Then, we introduce the proposed bimanual gesture recognition approach and details of our fusion process in Section 4. Experiments and comparison to existing methods are presented in Section 5. Finally, we demonstrate two multimedia applications in Section 5.5 and conclude in Section 6.

2 Related works

Based on the technique, hand gesture recognition methods can be mainly classified into glove-based, colored marker, and vision-based approaches.

Glove-based methods employ special designed glove with electromechanical parts to know the relative position of various hand sections of user [14, 24]. Its accuracy in measuring hand gesture is supposed to be the highest among the three approaches. However, the needs of special instrument hinder glove-based methods to be popular.

Colored marker approaches [39] thus require users to paint their hands or wearing gloves with colors in different sections. The orientation and pose of hands can be estimated quickly by the colors. However, the criteria of color painted hands look weird and definitely less attractive then vision-based recognition of bare hands.

Comparing to these two approaches, vision-based methods do not require any special device other than cameras. Erol et al. [9], Sarkar et al. [31] as well as Rautaray and Agrawal [28] had presented detailed reviews about vision-based hand tracking. Later, vision-based recognition are further enhanced by hand extraction using depth sensing techniques with Kinect [2] or Leap Motion [25]. Poularakis and Katsavounidis [26], as well as Leite et al. [18] applied hand recognition with depth in VR or other environments. Wang et al. [38] locate and segment single hand from depth images acquired from Kinect. Their method then performs superpixels to represent the hand surfaces. EMD is applied to measure the distance between the distributions of the centers of superpixels. Tzionas et al. [36] further enhanced 3D hand tracking with discriminative salient points. More examples of depth-based hand recognition can be found in [6]. There are also attempts to augment optical based recognition with other modalities like in Chen et al. [4], they proposed to use inertial sensors attached on users to obtain motion for gesture analysis. They found a higher recognition rate with use of an HMM-based recognizer than the baseline approach using a statistical feature-based classifier.

Apart from using depth data, a number of recent works attempt to rely merely RGB images which have a wider availability. Chen et al. [5] choose to segment the hand into parts like palm and fingers. Then, according to the detected hand and its distance transform to a template, a rule classifier is applied to recognize the hand gestures. Bulugu and Ye [3] extracts the extended higher order local autocorrelation (HLAC) features from a log-polar transformed hand image. Linear Discriminant Analysis (LDA) is then applied to learn the extracted features and classify the gestures. Joshi et al. [13] first obtains the body skeleton in order to locate hand regions in the images. Then, a set of consecutive frames are converted to features using HOG and PCA, and feeds into a random forest classifier for the recognition of hand gestures. Zen et al. [41] proposed the Transductive Parameter Transfer which predicts a new personalized classifier for every user. In other words, they can automatically label the face or gesture images of a new user subject based on the learning results of other subjects. While their method focuses on recognizing the trajectory of hands instead of their pose, so it cannot be directly applied into our problem. Wu and Kang [40] segment the hands using dense optical flow. The fingers of the hand are segmented by calculating the average centroid distance of the contour. However, their method focuses on the fingertips for gesture recognition which is not reliable for bimanual gesture as fingers are often being occluded. Despite all these methods work well for classic RGB images, they are designed and tested for recognizing single hand gestures only.

Another branch of attempts tried to solve the recognition with deep convolutional neural network which had been used to tackle many challenging vision-based recognition tasks. Tang et al. [34] employs straight forwardly the existing LeNet [17] model for hand gesture datasets. Barros et al. [1] and Molchanov et al. [20] incorperate 3D convolution to handle sequence of RGB-D hand gesture images. Molchanov et al. [21] extends their previous work with a recurrent 3D convolution network so that simultaneous detection and classification of hand can be achieved. It is worth to note that all the above methods use only single RGB or depth sensor which do not nicely handle the problem of occlusion.

However, most of existing work only focus on single-hand recognition. Recently, a few attempts are targeting for bimanual gesture recognition. Kristensson et al. [16] relies on the more robust fully-body skeleton recognition in identifying hand positions, and then enhances the two-hand motional gesture with probabilistic algorithm. However, the work does not put much efforts in recognizing the pose of hand. Oikonomidis et al. [23] proposes recognizing two hands using RGB-D camera with a model-based method that matches all the reprojected hand gestures with the use of Particle Swan Optimization (PSO). However, the tracking fails when there is self-occlusion of two hands. Schramm et al. [32] proposed a method to match music conducting gestures with the purpose of learning and evaluations. Their work acquired RGB-D frames with two arms' motions and use a modified DWT statistical classifier to estimate the temporal accuracy of the beat done by the music conductor. Saeed et al. [30] improves bimanual tracking with temporal coherence and assume that the pose of the hand does not change when being occluded, which is not always reasonable.

The works by Deng et al. [7] and Ge et al. [11] both employed 3D convolution in their neural network, but they first convert the depth map to a 3D volumetric representation before feeding it into the network to achieve better recognition results. Wan et al. [37] used two generative neural networks in order to learn the coupling and mapping between hand pose and corresponding depth image, thus they call it CrossNet which involves training a number of networks in complex steps. Recently, Mueller et al. [22] presented a method using two subsequently applied CNNs to overcome clutter and occlusions. Their method focused on clutter appeared in hand-object interaction instead of hand-hand interaction as ours. Thus, they assumed single hand will appear with depth (RGB-D) which is different from our goal in recognizing two hands from RGB data. All these methods relied on depth information from the image to obtain accurate results.

In contrast, our method tries to solve the occlusion problem with only RGB images but in a multi-view approach. By observing from a wider range of viewing angles to the two hands, we increase the chance to avoid confusion made by occlusion, and therefore improve our confidence in identifying a certain hand posture.

3 Multi-view setup

Before going into the detail of our method, we first introduce our multi-view setup for acquiring hand gesture data. One major difficulty in recognizing bimanual gestures is its highly self-occluded properties. Figure 3 shows two examples of bimanual gestures. These two gestures look rather different from the front view, while we cannot easily differentiate them from the side view. It is because the unique features of these gestures are being

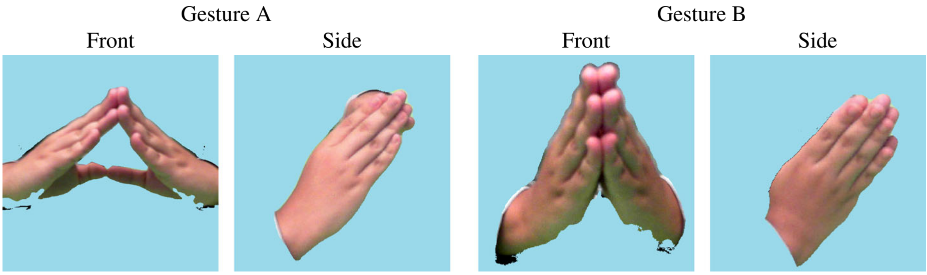


Fig. 3 Gestures may look different in certain view, but similar in another view due to self-occlusion

occluded. This example illustrates that bimanual gestures are prone to ambiguity if we recognize the gestures with only one view. To overcome this problem, a direct solution is to employ multi-view approach in order to minimize the effect of self-occlusion.

Figure 4 shows our multi-view setup. Three cameras are installed in diversified viewing points. We named the views of these three cameras as left, front, and right view respectively. It is worth to note that our setup does not require precise positioning and orientation setting during installation, a rough one is enough. To calibrate the camera, we place a marker on the capturing area, and roughly adjust the cameras until the marker appears in the same position on the screen for all experiments. Thanks to the robustness of our learning-based method, our recognition is highly flexible and repeatable for different environments.

4 Bimanual gesture recognition

By using our tri-camera setup, we captured a set of bimanual gesture images(examples in Fig. 5) from three different views. Our training samples are prepared by labeling a square hand region inside each camera frame, followed by rescaling and feature extraction. Finally, we employ Support Vector Machine (SVM), which is widely used for recognition, to learn a robust classification of the hand features.

4.1 Image features extraction

To improve robustness in learning, it is common to extract features from the images. Similar to most of the existing hand recognition methods, our method (Fig. 6) begins with a segmentation of hand based on the skin color. It allows the classifier to ignore most of the background, and focus on the targeted hands in the image. We do this by employing the skin color model of [27], which is a set of color criteria for skin in RGB, HSV, and YCrCb spaces. The use of multiple color spaces suppresses the effect of different lighting conditions, mainly addresses the lighting direction and intensity variations. Those criteria are developed based on learning from large number of cases. While by observing the criteria of

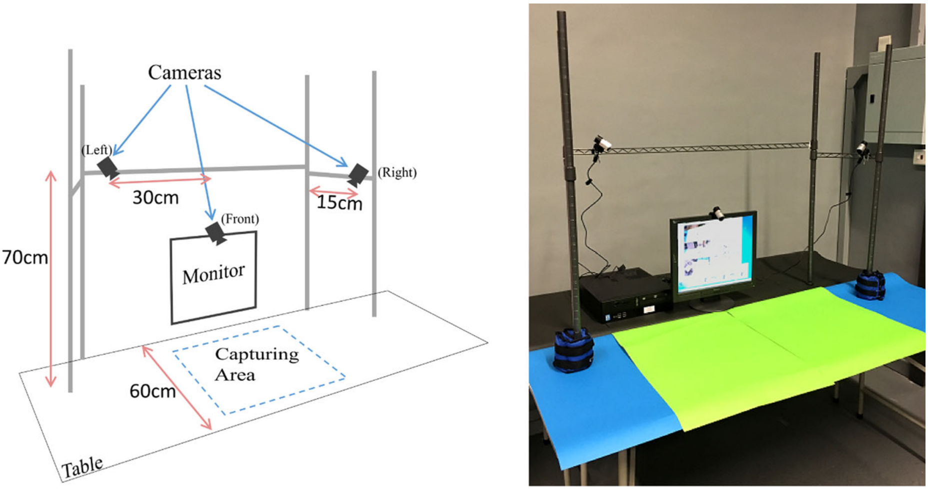
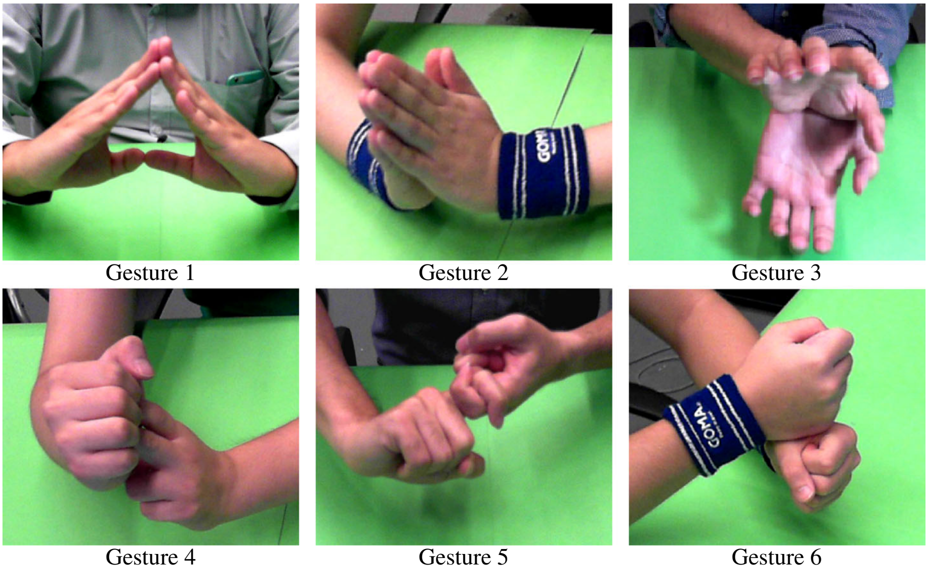


Fig. 4 Our multi-view setup environment with three cameras for capturing data


Gesture 1 Gesture 2 Gesture 3

Gesture 4 Gesture 5 Gesture 6

Fig. 5 The six bimanual gestures used in our experiments

their model [27], the chromaticity (e.g., CrCb and Hue) are major components taken into account in order to overcome the variance in illumination conditions.

With the segmented hand images, we normalize their size to 200 × 200 to obtain scale-invariant features. However, as the homogeneous and smooth skin surfaces, the variation of appearance of hands are relatively limited. Thus, we model the image features by considering its color and shape. Our color feature is constructed by computing the histogram of oriented gradients (HOG) in the color space. For shape feature, edge orientation histograms (EOH) after Canny edge detection is employed. To suppress the effect of lighting direction, we use unsigned orientation for both HOG and EOH in our model. Both HOG and EOH form feature vectors with 3600 dimensions.

Beside the color and shape feature, we had considered texture feature as well. We tried using Gabor filter [10], which is a prevalent approach for extracting useful texture features

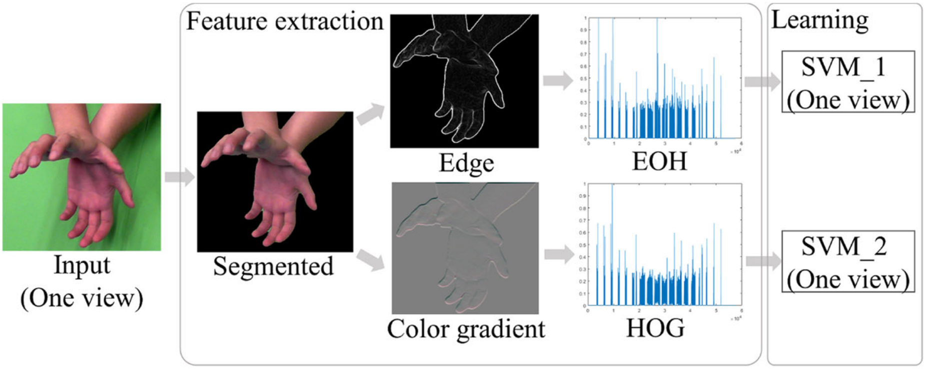


Fig. 6 The construction of features and learning process for a single view

in a scale and orientation invariant manner. As expected, the outcome is poor due to the fact that texture on smooth skin surface are relatively limited. The overall accuracy of it is less than 2% when applying to a SVM classification. Thus, we strongly believe texture feature is not helpful to the problem and is ignored in our bimanual recognition solution.

4.2 Ensemble of SVM classifiers

With the features extracted, we train the gesture classifier with SVM. However, naïvely putting all the features into single SVM is inappropriate in our multi-view situation, as the appearances of hands from different views vary significantly. Naïvely learning all features together greatly confuses the SVM classifier and leads to poor results.

Hence, we decide to tackle the different features from different views separately by utilizing independent SVM classifiers. Furthermore, two SVMs are used to handle color and shape features individually in order to minimize error introduced by the scale normalization of features.

In other words, each SVM is only trained with a single feature (Fig. 6) of one view. As a result, we used six different SVMs in total to learn the two features for the three views. In all SVMs, we use the Gaussian radial basis function as kernel with gamma $\gamma = 4.8$ to support non-linear learning.

These six trained SVMs can provide individual prediction based on corresponding input. To obtain a single answer out of these individual estimation, fusion method from classifier ensembling is employed. According to Dietterich [8] as well as Hansen and Salamon [12], a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse. In our case, as each classifier is either with a different input or a different feature extracted, it obviously leads to diversified individual.

To combine individual prediction which may be inconsistent to each other, a simple solution can be done by choosing one of the results with lowest classification score. However, some bimanual gestures may look similar in certain views and causes ambiguity (e.g. Fig. 3) when the unique features are occluded. Thus, we can expect a poor prediction will be resulted in such cases. Purely relying on one of the six results is therefore not a good idea in our method. We need to decide the most reliable result by considering all SVMs together.

Our idea is to fuse the results by a weighted sum on all SVM confidence scores from different views for different gestures. We have independent weightings for different gestures of different views. In other words, our 6-gesture and 3-view data will have 18 weightings in total. This idea is analogous to the work of Singha and Laskar [33], which developed a classifier fusion method for recognizing the trajectory of single hand movement. Figure 7 shows the flow of prediction of our method and the $\oplus$ symbols notate the weighted sum. The final score $S_g(I)$ of having gesture g in image I is defined as

$$S_g(I) = \omega_{g,L} \times S_{g,L}(I) + \omega_{g,F} \times S_{g,F}(I) + \omega_{g,R} \times S_{g,R}(I) \tag{1}$$

where L, F, and R represent the left, front, and right view respectively. ω is the salience of the view for different gesture. It is used to reduce the contribution of the view that may cause ambiguity in bimanual gesture. $S_{g,V}(I)$ is the confidence score from the SVM classifiers of view V, which is defined as weighted sum of SVM scores for color and shape features as follow:

$$S_{g,V}(I) = \alpha \times S_{g,V,color}(I) + (1 - \alpha) \times S_{g,V,shape}(I) \tag{2}$$

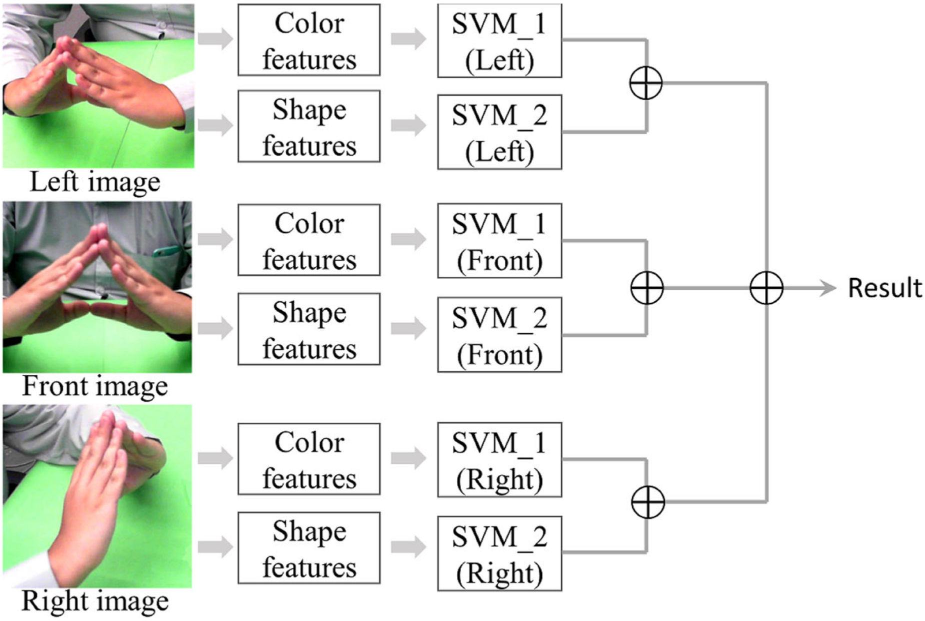


Fig. 7 Our prediction process. A final prediction is decided by combining six SVMs of different views and features

where α is the weighting in favor to color feature. In our experiments, we believe that both color and shape features are equally important. Thus, we set $\alpha = 0.5$ in all of our experiments. The final prediction result is the gesture g' with lowest final score $S'_g(I)$.

4.3 Optimization of weightings

To obtain the weighting ω for all gestures and views, we have the following assumption. If a gesture can be easily recognized by the images captured from a view (high accuracy), this gesture should have unique features in this view, so classification result from this view is more likely to be correct. In contrast, if a gesture is often classified wrongly in a view (low accuracy), certain ambiguity should have occurred, possibly because of occlusion. It would be better if we limit its contribution to the final result. Base on this assumption, to obtain the weightings, we first conduct a leave-one-out cross-validation (more detail in Section 5), and compute the accuracy of different gestures from different individual views (Table 2). The ratio of these accuracy values are used to calculate an initial ω of one view. Without the loss of generality, the initial weight of left view $\omega_{g,L}$ is defined as:

$$\omega_{g,L} = \frac{A_{g,L}}{A_{g,L} + A_{g,F} + A_{g,R}} \tag{3}$$

where A is the accuracy of recognition using single view. This equation also applies to right and front views.

Note that the initial weights here are not necessary to be optimal. Thus, we further adjust the weightings by optimization with simple gradient descent method and obtained the values of ω (Table 1). The objective function is simply the accuracy of using such weighting. For the negative data, we set the weightings of all the views equal.

Table 1 The weighting ω of views for different gestures

	Initial			Final		
	Left	Front	Right	Left	Front	Right
Gesture 1	0.28	0.46	0.26	0.00	0.92	0.08
Gesture 2	0.22	0.46	0.32	0.13	0.30	0.56
Gesture 3	0.35	0.32	0.33	0.04	0.28	0.67
Gesture 4	0.19	0.47	0.34	0.31	0.26	0.41
Gesture 5	0.32	0.38	0.30	0.30	0.61	0.07
Gesture 6	0.31	0.39	0.30	0.00	0.99	0.01
Negative	0.33	0.33	0.33	0.33	0.33	0.33

Our optimization starts with an initial weighting of the views (i.e., 3 different views in our case). Then, we adjust the weighting until the overall accuracy (4) improved and converged.

$$overall\_accuracy = \max\left(\frac{1}{N}\sum_{n=0}^{N}\frac{\left|I_n \cap \left\{x \in I \,|\, \arg\max(S_g(x)) = n\right\}\right|}{|I_n|}\right) \tag{4}$$

where N is the number of different gestures, in our case N is 6. I_n is the image of gesture n. $S_g(x)$ was defined in (1).

To avoid local minimum, we also randomly generate multiple initialization values as population and perform gradient descent multiple times. The instance with highest score is used as our final result. In other words, the final weighting should have the best accuracy among the case in our training data. Similar to all learning based methods, we believe that when the variety of training data is sufficiently large, these weights should be suitable for most cases.

5 Experimental results

We carried out a series of experiments to evaluate the performance of our bimanual gesture recognition method. Six subjects are invited to capture the gesture images by our tri-camera setup. To increase the variety of the data, these subjects are different in age and with different clothing (e.g., Long or short sleeve). We captured 10 s of gesture motion in 12 fps from each subjects for each gesture in one view. This results in a total of 2160 images for one gesture. For the sake of reproducibility, the whole dataset will be made available to public together with this paper. The Github link: https://github.com/geoffrey0822/handgesture_svm.

5.1 Accuracy analysis

Our first experiment is a leave-one-out cross-validation. This experiment is to validate the recognition performance based on separated training and testing samples. Each time we use the data set of one subject as a query, and the remaining data of other subjects as the training data set. The experiment is repeated for each subject. Table 2 shows the mean accuracy of recognition in this experiment.

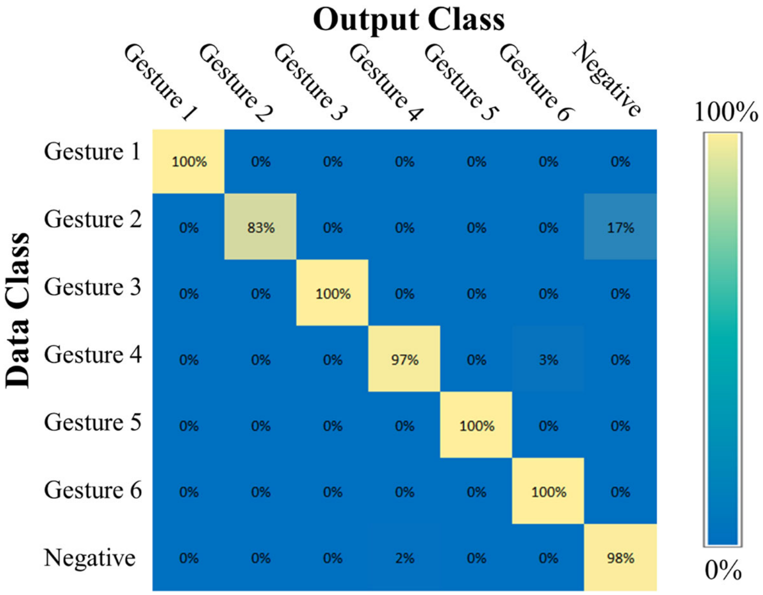
As expected, the front view classifier performed well for hand recognition comparing to other views'. It is because conventional hand gestures are often designed to show in front

Table 2 The accuracy of recognition using different views

	Left	Front	Right	Our method
Gesture 1	62.6%	100.0%	68.5%	100.0%
Gesture 2	55.6%	96.9%	59.3%	83.3%
Gesture 3	92.2%	100.0%	100.0%	100.0%
Gesture 4	58.1%	83.3%	70.5%	97.0%
Gesture 5	93.2%	98.1%	89.3%	100.0%
Gesture 6	91.3%	93.6%	82.1%	100.0%
Negative	96.9%	100.0%	95.3%	98.4%
Overall	78.5%	96.0%	82.1%	97.0%

of the other people for communication. While, we can still find the improved accuracy when combining three views with our method. From the table, it is obvious that most of our combined view results outperform other single view results. This validates the effectiveness of our multi-view approach.

Figure 8 shows the confusion matrix of our recognition using combined view. It indicates how many samples of a data class (Y-axis) is classified into an output class (X-axis). The population of each block is color-coded from blue (Less samples) to yellow (More samples). From the confusion matrix, there is an obvious diagonal highlighted in yellow, which means that most of the samples can be classified into the correct classes. It evidences that our method works well for bimanual gesture recognition.



	Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5	Gesture 6	Negative
Gesture 1	100%	0%	0%	0%	0%	0%	0%
Gesture 2	0%	83%	0%	0%	0%	0%	17%
Gesture 3	0%	0%	100%	0%	0%	0%	0%
Gesture 4	0%	0%	0%	97%	0%	3%	0%
Gesture 5	0%	0%	0%	0%	100%	0%	0%
Gesture 6	0%	0%	0%	0%	0%	100%	0%
Negative	0%	0%	0%	2%	0%	0%	98%

Fig. 8 The confusion matrix of our method. Yellow represents more samples while blue represents less samples

5.2 Comparisons with existing methods

There is a number of existing methods which tackle the problem of hand recognitions in real time. We selected a few representatives based on their currency, used method and approaches, including Chen et al. [5] (**Segment-based**), Bulugu and Ye [3] (**HLAC+LDA**), Joshi et al. [13] (**HOGPCA+RF**), Barros et al. [1] (**Multichannel**) and Tang et al. [34] (**LeNet**). These methods are chosen because most of them involve only RGB data in their core algorithm and feature extraction procedures which are consistent to our input data. However, unlike our method, all of these method are designed for single view input only. In order to have a fair comparison, we only apply the front view to these methods.

Figure 9 shows the experimental results of these methods together with our proposed one. Individual accuracy and overall accuracy are presented in the chart. Most of the existing methods do not perform well, having recognition accuracy below 50%, for our bimanual gesture data. We believe this is mainly due to the inability of feature extraction approach used in capturing the unique characteristics of bimanual gestures. Each method also has its own limitations. For example, segment-based method assumes a front facing palm which is rare in bimanual gesture. The log-polar transformation in HLAC+LDA method causes a bias to the appearance near the center point of the palm, which is not reliable for bimanual gestures. The HOGPCA+RF method requires an accurate hand region extraction from body skeleton, as well as an over reduced dimensionality with PCA in the feature; all of these introduce difficulties in properly recognizing complicated bimanual gestures. Only LeNet is comparable to our method, and has a high accuracy over 70% for all cases. Despite the LeNet method achieved a rather satisfactory result, our proposed framework still outperforms it in most of the gestures as well as the overall accuracy. As convolutional neural network optimizes proper feature itself in the training process, its major difference to our method becomes the multi-view inputs. It is very likely that our multi-view approach provides better tolerant to the occlusion appeared in gestures and improves the recognition to a certain extent.

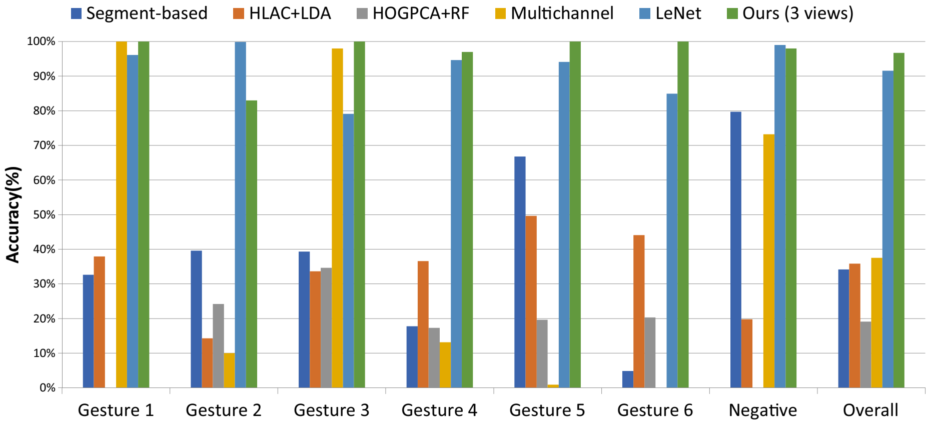


Fig. 9 The recognition accuracy of our method and existing ones including Segment-based, LDA, HOG-PCA+RF, Multichannel and LeNet

Table 3 Rate of recall and precision of selected fusion methods. Results from individual views (front, left, and right) are shown at the bottom

<table>
<tr><th>Fusion method</th><th colspan="8">Rate of recall | precision</th></tr>
<tr><th></th><th>Gesture1</th><th>Gesture2</th><th>Gesture3</th><th>Gesture4</th><th>Gesture5</th><th>Gesture6</th><th>Negative</th><th>Overall</th></tr>
<tr><td>Min</td><td>1.00|1.00</td><td>0.76|1.00</td><td>1.00|1.00</td><td>0.89|0.99</td><td>0.99|0.96</td><td>0.99|0.85</td><td>0.95|0.34</td><td>0.940|0.877</td></tr>
<tr><td>Sum</td><td>0.98|0.99</td><td>0.75|1.00</td><td>1.00|1.00</td><td>0.93|1.00</td><td>0.96|0.95</td><td>0.99|0.85</td><td>0.97|0.34</td><td>0.940|0.876</td></tr>
<tr><td>Product</td><td>0.97|0.99</td><td>0.82|1.00</td><td>1.00|1.00</td><td>0.93|1.00</td><td>0.95|0.93</td><td>0.99|0.90</td><td>0.97|0.34</td><td>0.946|0.881</td></tr>
<tr><td>Weighted min</td><td>1.00|0.85</td><td>0.48|1.00</td><td>0.43|1.00</td><td>0.00|0.00</td><td>0.60|0.96</td><td>1.00|0.33</td><td>0.03|0.02</td><td>0.506|0.592</td></tr>
<tr><td>Weighted sum</td><td>1.00|1.00</td><td>0.83|1.00</td><td>1.00|1.00</td><td>0.97|1.00</td><td>1.00|1.00</td><td>1.00|0.97</td><td>0.98|0.34</td><td><b>0.970</b>|<b>0.902</b></td></tr>
<tr><td>Voting</td><td>0.84|0.93</td><td>0.61|0.94</td><td>1.00|1.00</td><td>0.94|1.00</td><td>0.97|0.85</td><td>1.00|0.81</td><td>0.98|0.32</td><td>0.907|0.835</td></tr>
<tr><td>Norm-2</td><td>0.98|0.99</td><td>0.68|1.00</td><td>1.00|1.00</td><td>0.94|1.00</td><td>0.98|0.95</td><td>0.99|0.81</td><td>0.97|0.34</td><td>0.933|0.871</td></tr>
<tr><td>Front only</td><td>1.00|1.00</td><td>0.83|1.00</td><td>0.98|1.00</td><td>0.94|1.00</td><td>1.00|0.96</td><td>1.00|0.95</td><td>0.97|0.34</td><td>0.960|0.892</td></tr>
<tr><td>Left only</td><td>0.63|1.00</td><td>0.56|0.93</td><td>0.92|1.00</td><td>0.58|0.90</td><td>0.93|0.69</td><td>0.91|0.52</td><td>0.97|0.26</td><td>0.785|0.757</td></tr>
<tr><td>Right only</td><td>0.68|0.85</td><td>0.59|0.64</td><td>1.00|1.00</td><td>0.71|0.94</td><td>0.89|0.70</td><td>0.82|0.74</td><td>0.95|0.27</td><td>0.807|0.733</td></tr>
</table>

Highest among all other fusion methods are highlighted in bold

5.3 Different fusion schemes

Our employment of various views facilitate occlusion handling of bimanual gestures, but it also requires to combine results from multiple classifiers. This experiment tries to investigate the best fusion scheme so as to obtain a unified classification result with highest accuracy. Based on the works of Kittler et al. [15] and Tax et al. [35], we tested the common schemes of score-based classifier fusion including simple summation, product, minimum/maximum, weighted sum, weighted min/max and voting. For the weighted sum and weighted min methods, we compute all the weights by optimization as in Section 4.3.

Table 3 summarized the rate of recall and precisions for all different gestures of our data and the overall result. Most of them obtain better results than any of the single view (see front, left and right results at bottom of the table). Among all of the fusion methods, weighted sum achieves the best in both overall rates in recall and precision which are 0.970 and 0.902 respectively.

Figure 10 extracts a few samples from different gestures to allow a better understanding of the fusion and its effectiveness. On top right corner of each frame lists the classification scores of the frame to a certain gesture (e.g. gesture 1 is indicated with G1, gesture 2 with G2, and so on). This score represents the distance to the class boundary in SVM, therefore, a lower score means a higher chance to belong to that class. In most of the cases, the predictions from different views are correct and consistent to each other, such as gesture 2 and 3. These simple cases will always receive a fused prediction the same as all views. While there are cases that inconsistency occurs in the prediction among views.

From the case of gesture 1 (first row in Fig. 10), we can find that the lowest classification score in left and right views goes to gesture 5, while front view predicts gesture 1. This conflict is solved by our optimized fusion scheme mentioned, and a final score is computed as shown on the right. In this case, gesture 1 received the lowest final score 0.26 which is consistent to the ground truth. Case of gesture 4 also has similar situation, where both left and front views predict wrongly, but right view obtains a correct prediction. The fusion process then based on the optimized weighting to finalize the prediction as gesture 4 which is the right answer.

5.4 Analysis of features and classifiers

We choose to use color and shape as the features for gesture recognition. Their abilities can be observed from the recognition accuracy in Fig. 11. At the same time, we are curious on the performance of different classification methods. Thus, experiments are carried out with different combinations of features and classifiers which included popular ones LDA [3], Random Forest [13] and SVM (ours) classifiers. This experiment uses multi-view data and repeats the fusion procedures as in Section 4.3 but with a different classifier and features. The results are plotted in Fig. 11, and one can easily find that our proposed solution achieves the best result. The shape feature (EOH) does not fit well to the LDA classifier, and thus leads to poor results for both EOH and HOG+EOH features in LDA. However, using both color and shape features (HOG+EOH) for SVM can yield the highest recognition rate. Although purely using color features (HOG) alone with SVM can also yield a high enough recognition rate, it does not perform well for some individual bimanual gestures. This is also the reason of choosing color and shape with the SVM classifier in our proposed method.

5.5 Application and comparison with existing solutions

We propose two potential applications in great demand of bimanual inputs. The first application serves for gaming purpose for controls. We assign two of our gestures, gesture 1 and 3 in Fig. 5, with different meanings for gaming such as "defense" and "attack" respectively. This allows us to create an interactive combat game using these gestures. The second application is a sign language learner. The gestures 5 and 6 in Fig. 5 are sign language. They represent "friends" and "coffee" respectively.

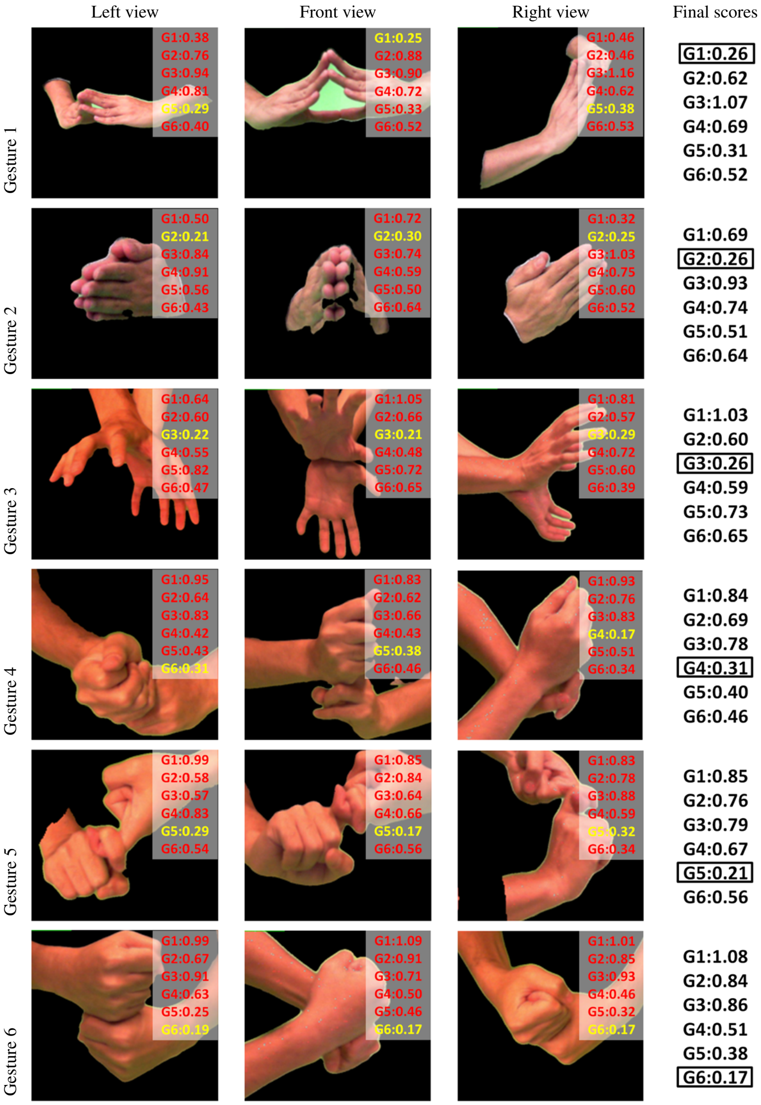

Left view
Front view
Right view
Final scores
Gesture 1
G1:0.38 G2:0.76 G3:0.94 G4:0.81 G5:0.29 G6:0.40
G1:0.25 G2:0.88 G3:0.90 G4:0.72 G5:0.33 G6:0.52
G1:0.46 G2:0.46 G3:1.16 G4:0.62 G5:0.38 G6:0.53
G1:0.26 G2:0.62 G3:1.07 G4:0.69 G5:0.31 G6:0.52
Gesture 2
G1:0.50 G2:0.21 G3:0.84 G4:0.91 G5:0.56 G6:0.43
G1:0.72 G2:0.30 G3:0.74 G4:0.59 G5:0.50 G6:0.64
G1:0.32 G2:0.25 G3:1.03 G4:0.75 G5:0.60 G6:0.52
G1:0.69 G2:0.26 G3:0.93 G4:0.74 G5:0.51 G6:0.64
Gesture 3
G1:0.64 G2:0.60 G3:0.22 G4:0.55 G5:0.82 G6:0.47
G1:1.05 G2:0.66 G3:0.21 G4:0.48 G5:0.72 G6:0.65
G1:0.81 G2:0.57 G3:0.29 G4:0.72 G5:0.60 G6:0.39
G1:1.03 G2:0.60 G3:0.26 G4:0.59 G5:0.73 G6:0.65
Gesture 4
G1:0.95 G2:0.64 G3:0.83 G4:0.42 G5:0.43 G6:0.31
G1:0.83 G2:0.62 G3:0.66 G4:0.43 G5:0.38 G6:0.46
G1:0.93 G2:0.76 G3:0.83 G4:0.17 G5:0.51 G6:0.34
G1:0.84 G2:0.69 G3:0.78 G4:0.31 G5:0.40 G6:0.46
Gesture 5
G1:0.99 G2:0.58 G3:0.57 G4:0.83 G5:0.29 G6:0.54
G1:0.85 G2:0.84 G3:0.64 G4:0.66 G5:0.17 G6:0.56
G1:0.83 G2:0.78 G3:0.88 G4:0.59 G5:0.32 G6:0.34
G1:0.85 G2:0.76 G3:0.79 G4:0.67 G5:0.21 G6:0.56
Gesture 6
G1:0.99 G2:0.67 G3:0.91 G4:0.63 G5:0.25 G6:0.19
G1:1.09 G2:0.91 G3:0.71 G4:0.50 G5:0.46 G6:0.17
G1:1.01 G2:0.85 G3:0.93 G4:0.46 G5:0.32 G6:0.17
G1:1.08 G2:0.84 G3:0.86 G4:0.51 G5:0.38 G6:0.17

Fig. 10 Extracted sample frames from various bimanual gestures. Classification scores of each frame are shown on the top right corners. The final scores are obtained by fusing of scores from different views

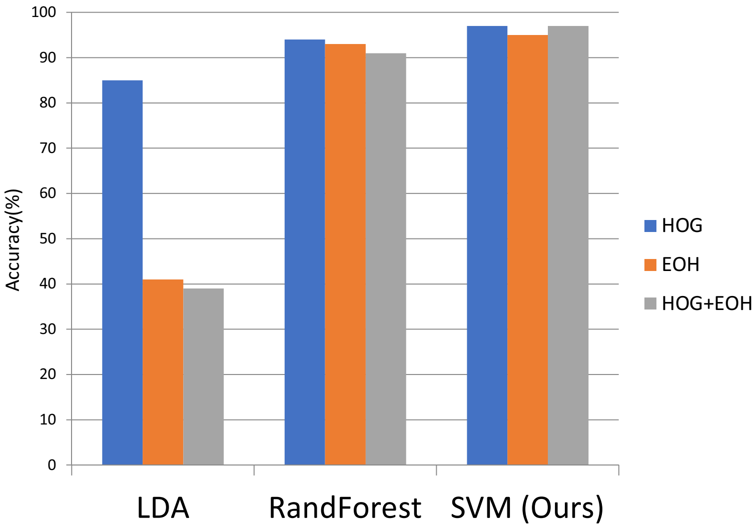
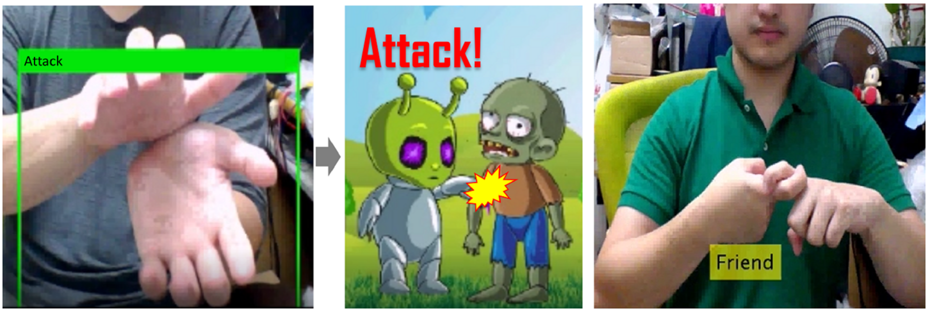


Fig. 11 The comparison of performance for various features and classifiers

If we implement the recognition of these bimanual gesture with existing commercial sensors like Kinect [2, 29] and Leap Motion [25] which are widely used for gesture recognition. It will easily suffer from poor results. These two sensors required to reconstruct the skeleton model of our body or hand, which often makes an assumption that every part of our body or hand is visible to the sensor. Thus, when the hands are highly overlapped, their skeleton reconstruction is error-prone and fail to recognize the hand gesture properly.

We do the comparison by performing the same gestures as in Fig. 5 and try to recognize with single Kinect and single Leap Motion. According to our experiment, Kinect can only achieve 60% accuracy for bimanual gesture. Leap Motion even failed to detect the existence of hands due to the self-occlusion.

In contrast, our gesture recognition enables the development of both interactive game and learning system for people to self-learning sign languages (Fig. 12). Readers are recommended to refer to our accompanied video for the interactive-rate performance of these two applications.



(a) Gamin g (b) Sign language learner

Fig. 12 The potential applications of our methods

6 Conclusions and future works

We presented a learning-based method to perform bimanual gesture recognition using SVM. To tackle the self-occlusion problem, we use multiple cameras from different viewing angles to capture the hand data. Hence, ambiguity occurs in one view can be compensated by another view. To align the results from different views, a weighted sum fusion scheme is proposed. Optimal weightings are obtained in order to adjust the contribution of a particular view based on its ambiguity on a certain gesture.

While our current method has several limitations. First, we assume the input data are static images. Thus, our method cannot handle gestures with motion well in the current implementation. In the future, we will extend our method to motion gestures by using multiple consecutive frames as learning data. Second, our segmentation purely based-on the skin color. Similar to others, using skin color may not be reliable. Lastly, we assume that the hand regions in the data are manually labeled on the image. However, we believe that this problem can be easily solved by any hand tracking or hand detection methods. But more investigations are needed.

Our current method focuses on a multiple view analysis of the bimanual gesture recognition. To the best of our knowledge, there does not exist similar datasets for training and testing. In the future, we are preparing to put more effort in collecting a larger number of bimanual hand samples. Last but not least, we are also preparing to apply deep learning approaches for this problem.

Acknowledgments The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Ref. No. UGC/FDS11/E03/15).

References

1. Barros P, Magg S, Weber C, Wermter S (2014) A multichannel convolutional neural network for hand posture recognition. Springer, Berlin, pp 403–410
2. Biswas KK, Basu SK (2011) Gesture recognition using microsoft kinect®. In: 5Th international conference on automation, robotics and applications. IEEE, ICARA, pp 100-103
3. Bulugu I, Ye Z (2016) Scale invariant static hand-postures detection using extended higher-order local autocorrelation features. Int J Comput Appl 135(5):1–5. published by Foundation of Computer Science (FCS), NY, USA
4. Chen M, AlRegib G, Juang BH (2013) Feature processing and modeling for 6d motion gesture recognition. IEEE Trans Multimedia 15(3):561–571. https://doi.org/10.1109/TMM.2012.2237024
5. hua Chen Z, Kim JT, Liang J, Zhang J, Yuan YB (2014) Real-time hand gesture recognition using finger segmentation. The Scientific World Journal 2014, Article ID 267872
6. Cheng H, Yang L, Liu Z (2016) Survey on 3d hand gesture recognition. IEEE Trans Circuits Syst Video Technol 26(9):1659–1673. https://doi.org/10.1109/TCSVT.2015.2469551
7. Deng X, Yang S, Zhang Y, Tan P, Chang L, Wang H (2017) Hand3D: hand pose estimation using 3d neural network. CoRR arXiv:1704.02224[cs.CV]
8. Dietterich TG (2000) Ensemble methods in machine learning. In: Proceedings of the first international workshop on multiple classifier systems, Springer-Verlag, London, UK, UK, MCS '00, pp 1–15, http://dl.acm.org/citation.cfm?id=648054.743935
9. Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: a review. Comput Vis Image Underst 108(1-2):52–73
10. Fogel I, Sagi D (1989) Gabor filters as texture discriminator. Biol Cybern 61(2):103–113
11. Ge L, Liang H, Yuan J, Thalmann D (2017) 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: Proc CVPR
12. Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001. https://doi.org/10.1109/34.58871

13. Joshi A, Monnier C, Betke M, Sclaroff S (2015) A random forest approach to segmenting and classifying gestures. In: 2015 11Th IEEE international conference and workshops on automatic face and gesture recognition (FG), vol 1, pp 1–7
14. Karime A, Al-Osman H, Gueaieb W, Saddik AE (2011) E-glove: an electronic glove with vibro-tactile feedback for wrist rehabilitation of post-stroke patients. In: IEEE international conference on multimedia and expo
15. Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20(3):226–239
16. Kristensson PO, Nicholson T, Quigley A (2012) Continuous recognition of one-handed and two-handed gestures using 3d full-body motion tracking sensors. In: Proceedings of the 2012 ACM international conference on intelligent user interfaces. ACM, pp 89-92
17. LeCun Y, Bengio Y (1998) The handbook of brain theory and neural networks. In: MIT Press, Cambridge, MA, USA, chap convolutional networks for images, speech, and time series, pp 255–258, http://dl.acm.org/citation.cfm?id=303568.303704
18. Leite DQ, Duarte JC, Neves LP, de Oliveira JC, Giraldi GA (2016) Hand gesture recognition from depth and infrared kinect data for cave applications interaction. Multimed Tools Appl:1–33
19. Li M, Leung H (2016) Multiview skeletal interaction recognition using active joint interaction graph. IEEE Trans Multimedia 18(11):2293–2302. https://doi.org/10.1109/TMM.2016.2614228
20. Molchanov P, Gupta S, Kim K, Kautz J (2015) Hand gesture recognition with 3d convolutional neural networks. In: 2015 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 1–7. https://doi.org/10.1109/CVPRW.2015.7301342
21. Molchanov P, Yang X, Gupta S, Kim K, Tyree S, Kautz J (2016) Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 4207–4215. https://doi.org/10.1109/CVPR.2016.456
22. Mueller F, Mehta D, Sotnychenko O, Sridhar S, Casas D, Theobalt C (2017) Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: Proc ICCV, http://handtracker.mpi-inf.mpg.de/projects/OccludedHands/
23. Oikonomidis I, Kyriazis N, Argyros AA (2012) Tracking the articulated motion of two strongly interacting hands. In: CVPR, IEEE computer society, pp 1862–1869
24. Parvini F, Mcleod D, Shahabi C, Navai B, Zali B, Ghandeharizadeh S (2009) An approach to glove-based gesture recognition. In: Proceedings of the 13th international conference on human-computer interaction. Springer-Verlag, Berlin, Heidelberg, pp 236–245, https://doi.org/10.1007/978-3-642-02577-8_26
25. Potter LE, Araullo J, Carter L (2013) The leap motion controller: a view on sign language. In: Proceedings of the 25th Australian computer-human interaction conference: augmentation, application, innovation, collaboration, ACM, pp 175–178
26. Poularakis S, Katsavounidis I (2014) Finger detection and hand posture recognition based on depth information. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 4329–4333, https://doi.org/10.1109/ICASSP.2014.6854419
27. Rahman N, Wei K, See J (2006) Rgb-h-cbcr skin colour model for human face detection. In: MMU international symposium on information & communications technologies (M2USIC 2006), MMU
28. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev 43(1):1–54
29. Ren Z, Yuan J, Meng J, Zhang Z (2013) Robust part-based hand gesture recognition using kinect sensor. IEEE Trans Multimed 15(5):1110–1120
30. Saeed A, Niese R, Al-Hamadi A, Michaelis B (2011) Coping with hand-hand overlapping in bimanual movements. In: IEEE international conference on signal and image processing applications. IEEE, ICSIPA, pp 238–243
31. Sarkar AR, Sanyal G, Majumder S (2013) Hand gesture recognition systems: a survey. Int J Comput Appl 71(15)
32. Schramm R, Jung CR, Miranda ER (2015) Dynamic time warping for music conducting gestures evaluation. IEEE Trans Multimedia 17(2):243–255. https://doi.org/10.1109/TMM.2014.2377553
33. Singha J, Laskar RH (2016) Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion. Multimedia Syst:1–16. https://doi.org/10.1007/s00530-016-0510-0
34. Tang A, Lu K, Wang Y, Huang J, Li H (2015) A real-time hand posture recognition system using deep neural networks. ACM Trans Intell Syst Technol 6(2):21:1–21:23
35. Tax DM, van Breukelen M, Duin RP, Kittler J (2000) Combining multiple classifiers by averaging or by multiplying? Pattern Recogn 33(9):1475–1485

36. Tzionas D, Ballan L, Srikantha A, Aponte P, Pollefeys M, Gall J (2016) Capturing hands in action using discriminative salient points and physics simulation. Int J Comput Vis 118(2):172–193. https://doi.org/10.1007/s11263-016-0895-4
37. Wan C, Probst T, Gool LV, Yao A (2017) Crossing nets: combining gans and vaes with a shared latent space for hand pose estimation. In: Proc CVPR
38. Wang C, Liu Z, Chan SC (2015) Superpixel-based hand gesture recognition with kinect depth camera. IEEE Trans Multimedia 17(1):29–39. https://doi.org/10.1109/TMM.2014.2374357
39. Wang RY, Popović J (2009) Real-time hand-tracking with a color glove. ACM Trans Graph 28(3):63:1–63:8
40. Wu G, Kang W (2016) Robust fingertip detection in a complex environment. IEEE Trans Multimedia 18(6):978–987. https://doi.org/10.1109/TMM.2016.2545401
41. Zen G, Porzi L, Sangineto E, Ricci E, Sebe N (2016) Learning personalized models for facial expression analysis and gesture recognition. IEEE Trans Multimedia 18(4):775–788. https://doi.org/10.1109/TMM.2016.2523421

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Geoffrey Poon is a research assistant at the School of Computing and Information Sciences in Caritas Institute of Higher Education, Hong Kong. He received his B.Sc., and M.Sc. degree from the Open University of Hong Kong and the Department of Computer Science and Engineering from The Chinese University of Hong Kong in 2013 and 2016 respectively. His research interests include deep learning, IoT applications, and robotics.

Kin Chung Kwan is a research fellow at the School of Computing and Information Sciences in Caritas Institute of Higher Education, Hong Kong. He received his B.Sc., and Ph.D. degree in the Department of Computer Science and Engineering from The Chinese University of Hong Kong in 2009 and 2015 respectively. His research interests include computer graphics, real-time rendering, nonphotorealistic rendering, GPGPU, and shape analysis.

Wai-Man Pang is now an Associate Professor cum Associate Dean at the School of Computing and Information Sciences in Caritas Institute of Higher Education, Hong Kong. He was with the Computer Graphics Lab, University of Aizu, Japan from 2009 to 2011 as an assistant professor. He finished his postdoctoral fellowship and Ph.D study at the Department of Computer Science and Engineering at CUHK.

His current research interests are two folds, one is on graphics related techniques and the other is healthcare related applications. Topics include texture analysis, vision based recognition, image feature extraction, computational manga, hardware accelerated algorithms and health care technologies on mobile devices.

During his academic career, he had presented and published many of his works on significant journal, books and conference in the related area, including ACM SIGGRAPH, ACM SIGGRAPH ASIA, ACM MM, IEEE TVCG, IEEE TMM, Sensors, and MICCAI. He had also served as program committee of IEEE ICSC, ICAT2E, ICHC, and etc, as well as reviewer of Siggraph Asia, Eurographics, CGF, Visual Computer and etc.