GESTURE-BASED USER AUTHENTICATION WITH KINECT

Liang Li

April 7 2013

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2013-2

BOSTON
UNIVERSITY

GESTURE-BASED USER AUTHENTICATION WITH KINECT

Liang Li

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

April 7 2013

Technical Report No. ECE-2013-2

Summary

Video cameras are extensively used in modern surveillance system to detect, track, and recognize, objects, people and anomalies. Their use in user authentication, however, has been limited primarily to close-range face recognition systems [2]. While, Kinect, in these cases, provides us a decent method to accomplish these goals. Kinect is a motion sensing input device by Microsoft, offering depth, RGB, and skeleton information. Moreover, Kinect sdk is open source, which is very user-friendly to all developers to design many products. However, kinect is rarely used for user authentication now. Thus, it is quite interesting to do research in this field. This master project explores user authentication based on gestures captured by a Kinect. Previous work on the same topic has been done in [2], which developed preliminary work. In paper [2], user authentication and gesture recognition experiments has been done based on our own dataset. In each kind of experiment, two tests has been conducted, one is false acceptance test, and the other is false rejection test. Equal error rate (EER) is defined to evaluated the experiment performance, which means that when false acceptance rate (FAR) equals to false rejection rate (FRR) In this master project, more thorough experiments has been developed on two datasets, one is our own dataset and another is Microsoft Kinect dataset. More detail conclusions are drawn at the end.

Keyword: *user authentication, gesture recognition, Kinect, FAR, FRR, EER*

Contents

List of Figures

List of Tables

1 Introduction

We have seen many authentication systems via face recognition at a close range. However, it's vulnerability to malicious attacks was demonstrated simply by placing a photo in front of a camera [2]. Another vulnerable authentication system example is a pure password system like your smart phone or personal computer. People can easily hack into this system if one peeps over your shoulder when you're entering password on keyboard.

Considering these issues, we come up with an idea that using biometric information in authentication system. There are, certainly, authentication systems via biometric information like fingerprint. But, once one hurts his fingers, one couldn't get access to this system. In contrast, human gait, human gesture and human signature are hardly to mimic. Gestures contain rich biometric information, can be repeatedly produced, and can be variously defined by different users. Thus if taking this kind of biometric information into authentication system, the authentication performance would be greatly enhanced. This master project develops research on human gesture based authentication.

There are two concepts of general authentication: recognition, and authentication. In this project, we define recognition as identification. **In identification level**, one provides a gesture in front of Kinect. The system then compares the distance between this gesture and all the same kind of gestures in the dictionary. If there is at least one distance less than threshold θ, then this user is accepted. In other words, if all the distances are above the threshold θ, this user is rejected. **In authentication level**, one claims to be other user and provide a gesture. The system then compares the distance between this gesture and all the same kind of gestures only within the claimed user's gesture dictionary. If there is at least one distance less than threshold θ, then this user is accepted. If all the distances are above the threshold θ, this user is rejected. A more thorough discussion about the difference between recognition and authentication is provided in [2].

Two kinds of tests are developed to do identification and authentication, false acceptance test, and false rejection test. Gesture dataset is divided into unauthorized part and authorized part under all possible combinations. **In false acceptance test**, we test each gesture realization of unauthorized part against all same kind of gestures remained in the authorized part in **recognition case**, and test each gesture realization in the unauthorized part against all same kind of gestures of the claimed authorized users in **authentication case**. **In false rejection test**, one gesture realization is pulled out from authorized part, and test against all same kind of gestures remained in the authorized part in **recognition case**, and test against all same kind of gestures of the user himself in **authentication case**. Repeated for all gestures of all authorized users. Two parameters are achieved from false acceptance

test and false rejection test. **FAR**, false acceptance rate, and **FRR**, false rejection rate. The total number of false acceptances divided by the total number of tests is known as the False Acceptance Rate (FAR). This is repeated for various thresholds θ. Similarly, The total number of false rejections divided by the total number of tests is known as the False Rejection Rate (FRR). This is repeated for various thresholds θ. Finally, we have Equal Error rate (**EER**) when FAR equals to FRR. In this report, we evaluate the authentication performance by this EER.

2 Feature Covariance Matrix

Let $\mathcal{F} = \{\mathbf{f}_n, n = 1, ..., N\}$ denote a "bag of feature vectors" that generated from a gesture video. The empirical covariance matrix of $\mathcal{F}$ is defined by the following formula:

$$C = \frac{1}{N}\sum_{i=1}^{N}(f_n - \mu)^T(f_n - \mu) \tag{1}$$

where $\mu = \frac{1}{N}\sum_{i=1}^{N} f_n$ is the empirical mean feature vector.

If $\mathbf{f}_n$ is d dimension, then $\mathcal{C}$ is a $d \times d$ matrix. If the eigen-decomposition of $\mathcal{C}$ is given by $\mathcal{C} = VDV'$, where the columns of V are orthonormal eigenvectors and D is the diagonal matrix of (non-negative) eigenvalues, then $\log(\mathcal{C}) := V\tilde{D}V'$, where $\tilde{D}$ is a diagonal matrix which diagonal entries are the logarithms of corresponding entries of D. We define this $log(\mathcal{C})$ as log-covariance matrix. A distance between covariance matrices $\mathcal{Q}$ and $\mathcal{C}$ is calculated by the Euclidean distance between the two log- transformed representations:

$$\rho(\mathcal{Q}, \mathcal{C}) := \|log(\mathcal{Q}) - log(\mathcal{C})\|_2 \tag{2}$$

where $\|\cdot\|_2$ is the matrix Frobenius norm. Assume K denotes the number of users, and M denotes the number of the total kinds of gestures created by K users. We can generate a dictionary of covariance matrices $\{\mathcal{C}_1^1, \mathcal{C}_2^1, ..., \mathcal{C}_M^1, \mathcal{C}_1^2, ..., \mathcal{C}_M^2, ..., \mathcal{C}_1^K, ..., \mathcal{C}_M^K\}$. For a query gesture with log-covariance matrix $\mathcal{Q}$, the distances of $\mathcal{Q}$ and each covariance matrix $\mathcal{C}$ in the dictionary can be calculated as the following:

$$\begin{aligned} indiv(\mathcal{Q}) &= k^* \\ (k^*, m^*) &= arg \min_{\substack{k=1,...K \\ m=1,...,M}} \rho(\mathcal{Q}, \mathcal{C}_M^K) \end{aligned} \tag{3}$$

More thorough details about the derive of log-covarice matrix are provided in [2].

3 Gesture-based Features

Features are used to distinguish different gestures. In this project, features used to do authentication are generated according to two different information source provided by Kinect. One is silhouette sequences, and another is skeleton sequences.

3.1 Silhouette-based feature

Users perform gestures in front of the Kinect camera. The way to construct feature vectors is to base them on a sequence of 2D silhouettes. There are various background subtraction methods that can be used to estimate an object silhouette sequence from a raw video. Usually, binary silhouette are used when extracting gesture features, meaning that silhouette contains a white mask (set to 1) of the moving object, and black background (set to 0) [2]. Let (x, y, t) denote the horizontal, vertical and temporal coordinates of a pixel in a video sequence of a gesture. And each silhouette pixel are assigned an index, $n = 1, 2, \ldots, N$, so that N denotes the total number of pixels in the silhouette sequence. Then, a feature vector at (x,y,t) can be denoted as $f(x, y, t)$ or f_n. The following 13 dimensional feature vector are used to express the silhouette characteristics:

$$f_n := [x, y, t, d_E, d_w, d_N, d_S, d_{NE}, d_{SW}, d_{SE}, d_{NW}, d_{T+}, d_{T-}]^T, \tag{4}$$

where, d_E, d_W, d_N, and d_S are Euclidean distances from (x, y, t) to the nearest silhouette boundary point to the east, west, north, and south. Also, d_{NE}, d_{SW}, d_{SE}, and d_{NW} are Euclidean distances to the nearest boundary point in the four diagonal directions. Moreover, d_{T+} and d_{T-} are measurements in the temporal direction.

3.2 Skeleton-based Feature

Kinect also provides xyz coordinates of 20 skeleton joints which are head, shoulder center, shoulder left, shoulder right, elbow left, elbow right, wrist left, wrist right, hand left, hand right, spine, hip center, hip left, hip right, knee left, knee right, ankle left, ankle right, foot left and foot right. Typical skeleton frames are shown in Figure 1. For each skeleton frame, coordinates of 20 joints are put into one vector, we call this one feature. The following expression is an example of one feature vector:

$$f = [x_1, y_1, z_1, x_2, y_2, z_2, ..., x_{20}, y_{20}, z_{20}] \tag{5}$$

A complete gesture contains many skeleton frames, thus, one gesture have a set of feature vectors. For example, a complete gesture has N frames then this gesture has n feature vectors as the following:

$$\begin{aligned} f_1 &= [x_1, y_1, z_1, x_2, y_2, z_2, ..., x_{20}, y_{20}, z_{20}] \\ f_2 &= [x_1, y_1, z_1, x_2, y_2, z_2, ..., x_{20}, y_{20}, z_{20}] \\ &\vdots \\ f_n &= [x_1, y_1, z_1, x_2, y_2, z_2, ..., x_{20}, y_{20}, z_{20}] \end{aligned} \tag{6}$$

We can consider this set of feature vectors as a bag of features. We use this bag of features to achieve it's log-covariance matrix based on the method described in Chapter 2.

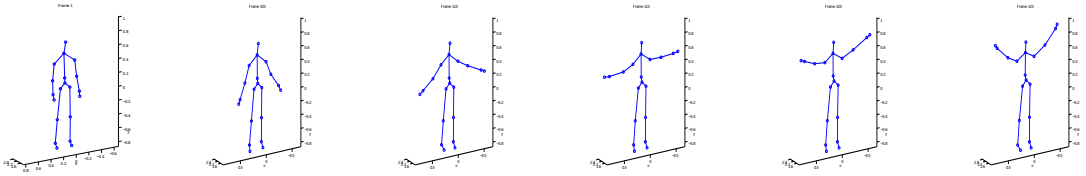

Figure 1: sequences of lift-both-arms gesture

4 Experiment and Results

Experiments are developed based on the proposed approach described above on two gesture datasets collected by Kinect. One dataset is collected by our own, and another dataset is provided by Microsoft Research Cambridge.

4.1 Dataset description

Our own dataset contains 8 gestures done by 20 users, and each gesture is performed 5 times by each user. The gestures are right-arm swing (to the left), left-arm swing (to the right), right-arm push, left-arm push, right-arm back, left-arm back, zoom-in (outward moving arms), zoom-out (inward moving arms). We collected both silhouette sequence and also skeleton sequence. Gestures are recorded at 30 fps. Each gesture realization lasts for 30 frames. Silhouette sequences and skeleton sequences are both collected. In total, we have $20 \times 8 \times 5$ gestures, and by 30 frames per realization we have 24,000 frames. More details about the dataset are presented in [2].

The Microsoft Research Cambridge-12 Kinect gesture dataset includes 12 gestures performed by 30 people. The motion files contain tracks of 20 joints estimated using the Kinect Pose Estimation pipeline. The body poses are captured at a sample rate of 30Hz. In one gesture video, one user performs one gesture repeatedly. In this situation, each gesture realization should be cut out from the whole video. We cut videos manually to ensure that one gesture realization is clear and complete. Therefore, we can't guarantee the same video length of each realization. Each realization lasts for at least 60 frames. In this master project, a subset of gestures are chose from the original dataset. This subdataset contains 5 gestures performed by 5 users, and 8 realizations for each gesture. In total, we have $5 \times 5 \times 8$ gestures, and by over 60 frames per realization we have at least 12,000 frames. The gestures are: kick, push-right(arm moving from left to right), bow, lift-both-arms, throw-object. The annotation of the start time and the end time of each gesture realization is given in the supplementary file. There are more details about this dataset in [1].

4.2 Baseline experiment description

In this project, two kinds of experiment are developed, user identification and user authentication. In both kinds of experiment, we divide K users into L unauthorized group and $(K - L)$ authorized group. We call this a $L/(K - L)$ split.

In identification test, given a specific threshold θ and split, each gesture realization in unauthorized group is pulled out and developed FA test against authorized group, also each gesture realization from authorized group is pulled out and developed FR test against authorized group. For example, total number of users are 20, and we

have 1/19 split. User1 is in the unauthorized group, while all the remaining 19 users are in the authorized group. User1 performs 1 specific gesture, like lift-both-arms, and this gesture is compared against all the lift-both-arms gestures of all 19 authorized users. The same test is developed repeatedly through different thresholds θ, different splits, and all the possible combinations within each specific split. Here all possible combinations means all possible combinations of dividing people into unauthorized group and authorized group. For our own dataset, since the number of total users are 20, it is impossible to finish tests of all possible combinations when L is large. In this cases, if L is larger than 5, we do random permutation for 6000 combinations. For Microsoft dataset, since total number of users is 5, we don't have this issue. All combinations are tested.

In authentication test, given a specific threshold θ and a specific split, each gesture realization in unauthorized group is pulled out and developed FAR test against authorized group, also each gesture realization from authorized group is pulled out and developed FRR test against authorized group. Also, here we give an example. Total number of users are 20, and we have 1/19 split. User1 is in the unauthorized group, while all the remaining 19 users are in the authorized group. User1 performs 1 specific gesture, like lift-both-arms, and he claims to be User10. This gesture is compared agains all the lift-both-arms gestures of User10. The same test is developed repeatedly through different thresholds θ , different splits, and all the possible combinations within each specific split. Again, we do random permutation for our own dataset when L is larger than 5. Figure 2 shows the process of experiment.

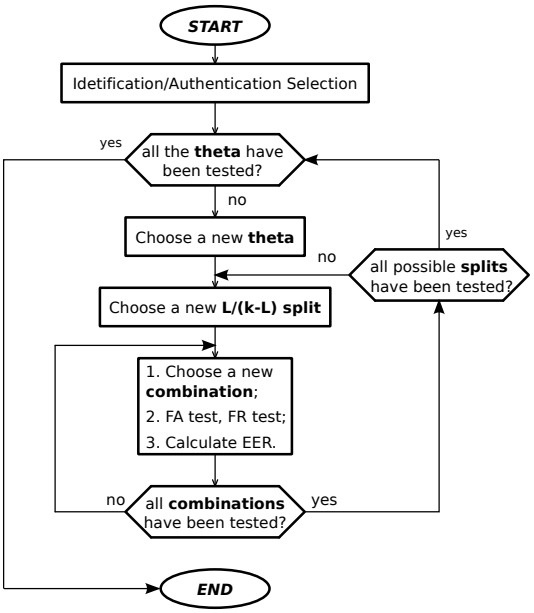

Figure 2: Diagram of experiment process

4.3 Baseline experiment results

Experiments are developed to test the proposed approach on two datasets. For our own dataset, we use silhouette sequences features as described in chapter 3 and also skeleton sequences features to do both identification and authentication tests. For Microsoft Kinect dataset, since it provides skeleton sequence only, we use skeleton features to do both identification and authentication tests.

4.3.1 User identification test

Table 1 shows the EER performance of user identification of silhouette sequence feature based on our own dataset. The tests are developed on different thresholds, different $L/(K-L)$ splits, and all possible combinations.

Table 1: User Identification EER Results (%) - Silhouette Feature

L/(K-L) Splits		(1/19)	(2/18)	(3/17)	(4/16)	(5/15)	(6/14)	(7/13)	(8/12)	(9/11)	(10/10)
Gesture	CL	10	9.8	9.7	9.55	9.4	9.2	9	8.8	8.42	7.93
	LB	0	0	0	0	0	0	0	0	0	0
	LP	4.7	4.6	4.5	4.4	4.3	4.15	4.05	3.9	3.71	3.58
	LS	4.5	4.4	4.2	4.15	4	4.02	3.93	4.01	4	4.02
	OP	11	10.5	9.95	9.4	8.8	8.4	8.01	7.91	7.83	7.73
	RB	0	0	0	0	0	0	0	0	0	0
	RP	8	8.1	8.1	8.2	8.2	8.3	8.34	8.37	8.53	8.51
	RS	7	7	7	7	7	6.9	6.72	6.51	6.3	6.13

Table 2 shows the EER performance of user identification of skeleton sequence feature based on our own dataset. The tests are developed on different thresholds, different $L/(K-L)$ splits, and all possible combinations.

Table 2: User Identification EER Results (%) - Skeleton Feature

L/(K-L) Splits		(1/19)	(2/18)	(3/17)	(4/16)	(5/15)	(6/14)	(7/13)	(8/12)	(9/11)	(10/10)
Gesture	CL	45.84	45.70	45.56	45.44	45.25	45.12	44.97	44.77	44.41	43.84
	LB	42.86	42.88	42.85	42.67	42.46	42.20	42.06	41.73	41.34	40.73
	LP	41.17	41.09	40.97	40.83	40.66	40.37	40.27	39.87	39.28	38.26
	LS	39.04	38.78	38.65	38.79	38.20	38.05	38.24	37.94	37.53	37.44
	OP	39.04	39.09	39.13	39.10	39.05	38.98	38.88	38.84	38.77	38.76
	RB	38.16	37.97	37.80	37.64	37.42	37.15	37.08	36.69	36.06	35.61
	RP	35.98	35.55	35.09	34.59	34.04	33.77	33.22	33.26	33.39	33.15
	RS	39.86	39.13	38.33	38.18	37.98	37.33	36.98	36.20	35.59	35.35

Table 3 shows the EER performance of user identification based on Microsoft kinect dataset. The tests are developed on different thresholds, different $L/(K-L)$ splits, and all possible combinations.

Table 3: User Identification EER Results (%) - Skeleton Feature

L/(K-L) Splits		(1/4)	(2/3)	(3/2)
Gesture	KICK	23.38	21.19	17.92
	PUSHRIGHT	2.55	2.57	2.56
	THROW	19.40	18.08	15.42
	BOW	18.37	18.78	14.39
	LIFT_BOTH_ARMS	9.19	8.86	8.20

4.3.2 User authentication test

Table 4 shows the EER performance of user authentication of silhouette sequence feature based on our own dataset.

Table 4: User Authentication EER Results(%) - Silhouette Feature

L/(K-L) Splits		(1/19)	(2/18)	(3/17)	(4/16)	(5/15)	(6/14)	(7/13)	(8/12)	(9/11)	(10/10)
Gesture	CL	2.48	2.48	2.48	2.48	2.48	2.46	2.46	2.49	2.47	2.48
	LB	0	0	0	0	0	0	0	0	0	0
	LP	2	2	2	2	2	1.99	2.02	1.99	1.97	1.98
	LS	2	2	2	2	2	2	1.98	2	1.98	1.99
	OP	8	8	8	8	7.96	7.97	8.02	7.96	7.93	7.88
	RB	0	0	0	0	0	0	0	0	0	0
	RP	3.28	3.29	3.29	3.28	3.28	3.28	3.28	3.3	3.29	3.29
	RS	3.9	3.9	3.89	3.9	3.9	3.89	3.9	3.89	3.89	3.91

Table 5 shows the EER performance of user authentication of skeleton sequence feature based on our own dataset.

Table 5: User Authentication EER Results (%) - Skeleton Feature

L/(K-L) Splits		(1/19)	(2/18)	(3/17)	(4/16)	(5/15)	(6/14)	(7/13)	(8/12)	(9/11)	(10/10)
Gesture	CL	24.42	24.42	24.42	24.42	24.42	24.47	24.47	24.36	24.41	24.50
	LB	21.58	21.58	21.58	21.58	21.58	21.60	21.61	21.57	21.63	21.59
	LP	17.25	17.25	17.25	17.25	17.25	17.19	17.27	17.27	17.29	17.23
	LS	25.43	25.43	25.43	25.43	25.43	25.46	25.49	25.41	25.37	25.61
	OP	25.49	25.49	25.49	25.49	25.49	25.53	25.50	25.49	25.50	25.54
	RB	18.16	18.16	18.16	18.16	18.16	18.12	18.22	18.19	18.18	18.24
	RP	20.34	20.34	20.34	20.34	20.34	20.43	20.41	20.30	20.39	20.37
	RS	18.43	18.43	18.43	18.43	18.43	18.42	18.44	18.45	18.39	18.41

Table 6 shows the EER performance of user authentication based on Microsoft kinect dataset.

4.4 Advanced experiment description

It is not surprising that in baseline experiments, results of silhouette-feature experiments are much better than skeleton-feature experiments. The reason is that there

Table 6: User Authentication EER Results (%) - Skeleton Feature

L/(K-L) Splits		(1/4)	(2/3)	(3/2)
Gesture	KICK	12.50	12.50	12.50
	PUSHRIGHT	2.50	2.50	2.50
	THROW	12.50	12.50	12.50
	BOW	12.50	12.50	12.50
	LIFT_BOTH_ARMS	7.50	7.50	7.50

are far less number of frames, or we can say samples, of one complete gesture than the dimension of the skeleton feature. Specifically, for our own dataset, one skeleton feature is 60 dimension, while single complete gesture only lasts 30 frames. Moreover, coordinates of skeleton joints are less robust since they can be ruined by twitching or occlusion. Figure 3 illustrates the case of occlusion when "bow" gesture is performed. It is possible that some joints are redundant, meaning that deleting the joints won't change the EER performance, also it is possible that certain joints are bad joints, meaning that using these joints makes EER performance worse. To solve this problem, we address a method of deleting several skeleton joints, and adding time reference to the skeleton feature.

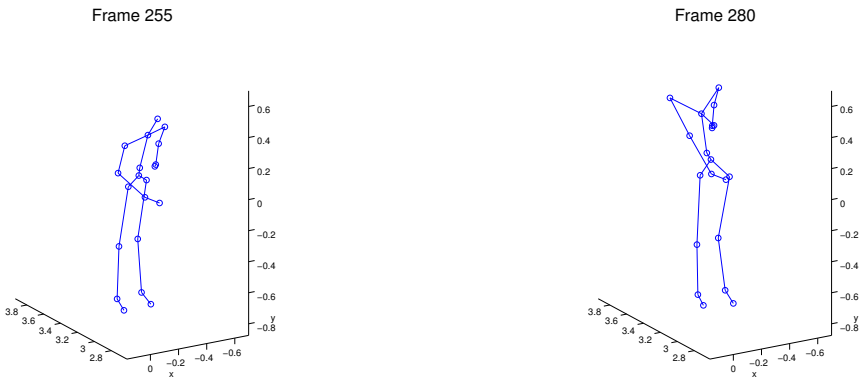

Figure 3: shows bow with occlusion

Based on our observation of the skeleton gesture video, we find out the following joints are twitching or going crazy the most. They are, left and right foot, hip center, spine, shoulder center. It is interesting to do test on deleting one or more of these crazy joints, adding time reference, and doing the mix combination, meaning that deleting one or more joints and adding time reference at the same time. There are 5 items in total, i.e, deleting left and foot joints, deleting hip center , deleting spine, deleting shoulder center, and adding time reference. Advanced authentication tests are developed through all possible combination of the 5 items. The total number of advanced tests equals to $\sum_{k=1}^{5} \begin{pmatrix} 5 \\ k \end{pmatrix}$, which is 31.

4.5 Advanced experiment results

In this part, the EER performance of 31tests for all advanced experiments, advanced user identification and advanced user authentication, are shown here. Since there are too many tests and different splits, we do not provide tables of results here. Instead, we prefer to plot the results graphically. Since EER results decrease as number of unauthorized user increases, results of split $L/(K-L)$ with the biggest L are plotted. For our own dataset, total number of users is 20, and we plot results of (10/10) splits only. For Microsoft dataset, the total number of users is 5, and we plot results of (3/2) splits only. The first data point in each plot indicates the EER results of original dataset feature, i.e baseline experiments. Table 7 shows the representation of X-axis indexes. Results of adding time reference and no-time reference are separated. X-index 2 to 17 are time-relevant, while X-index 18-32 are time-irrelevant.

Table 7: Explanation of 31 advanced experiments

Item					
X-axis index	add time reference	dlt left, right foot	dlt spine	dlt hip-center	dlt shoulder-center
1					
2	✓				
3	✓	✓			
4	✓		✓		
5	✓			✓	
6	✓				✓
7	✓	✓	✓		
8	✓	✓		✓	
9	✓	✓			✓
10	✓		✓	✓	
11	✓		✓		✓
12	✓			✓	✓
13	✓	✓	✓	✓	
14	✓	✓	✓		✓
15	✓	✓		✓	✓
16	✓		✓	✓	✓
17	✓	✓	✓	✓	✓
18		✓			
19			✓		
20				✓	
21					✓
22		✓	✓		
23		✓		✓	
24		✓			✓
25			✓	✓	
26			✓		✓
27				✓	✓
28		✓	✓	✓	
29		✓	✓		✓
30		✓		✓	✓
31			✓	✓	✓
32		✓	✓	✓	✓

4.5.1 Advanced user identification test

Figure 4 shows the results of 31 tests on single gesture of Microsoft kinect dataset in one plot.Figure 5 shows the results of 31 tests on each gesture of our own dataset.

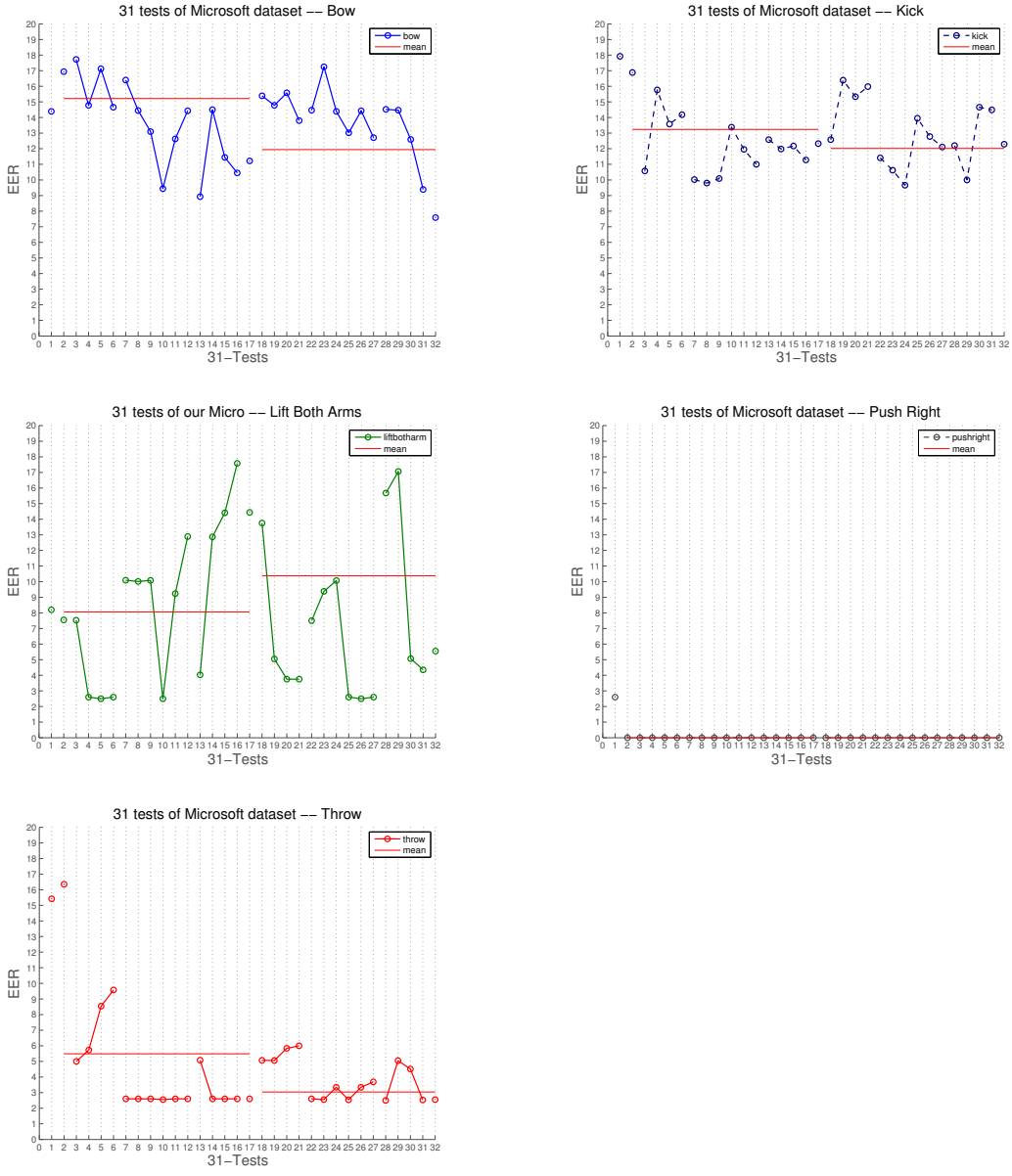

Figure 4: Results of EER for 31tests of User Identification on Microsoft dataset

4.5.2 Advanced user authentication test

Figure 6 shows the results of 31 tests on single gesture of Microsoft kinect dataset. Figure 7 shows the results of 31 tests on single gesture of our own dataset in one plot.

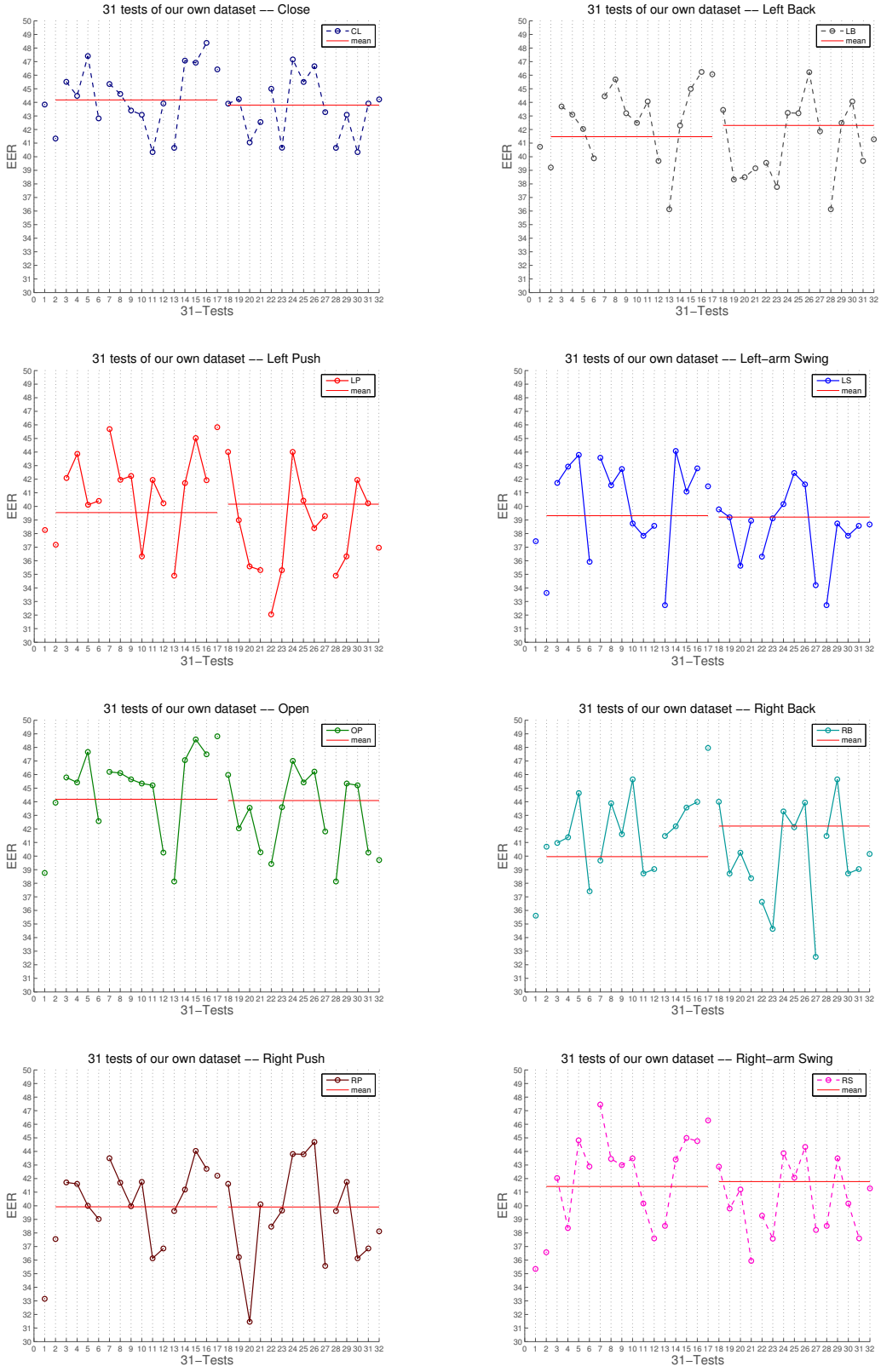

Figure 5: Results of EER for 31tests of User Identification on our own dataset

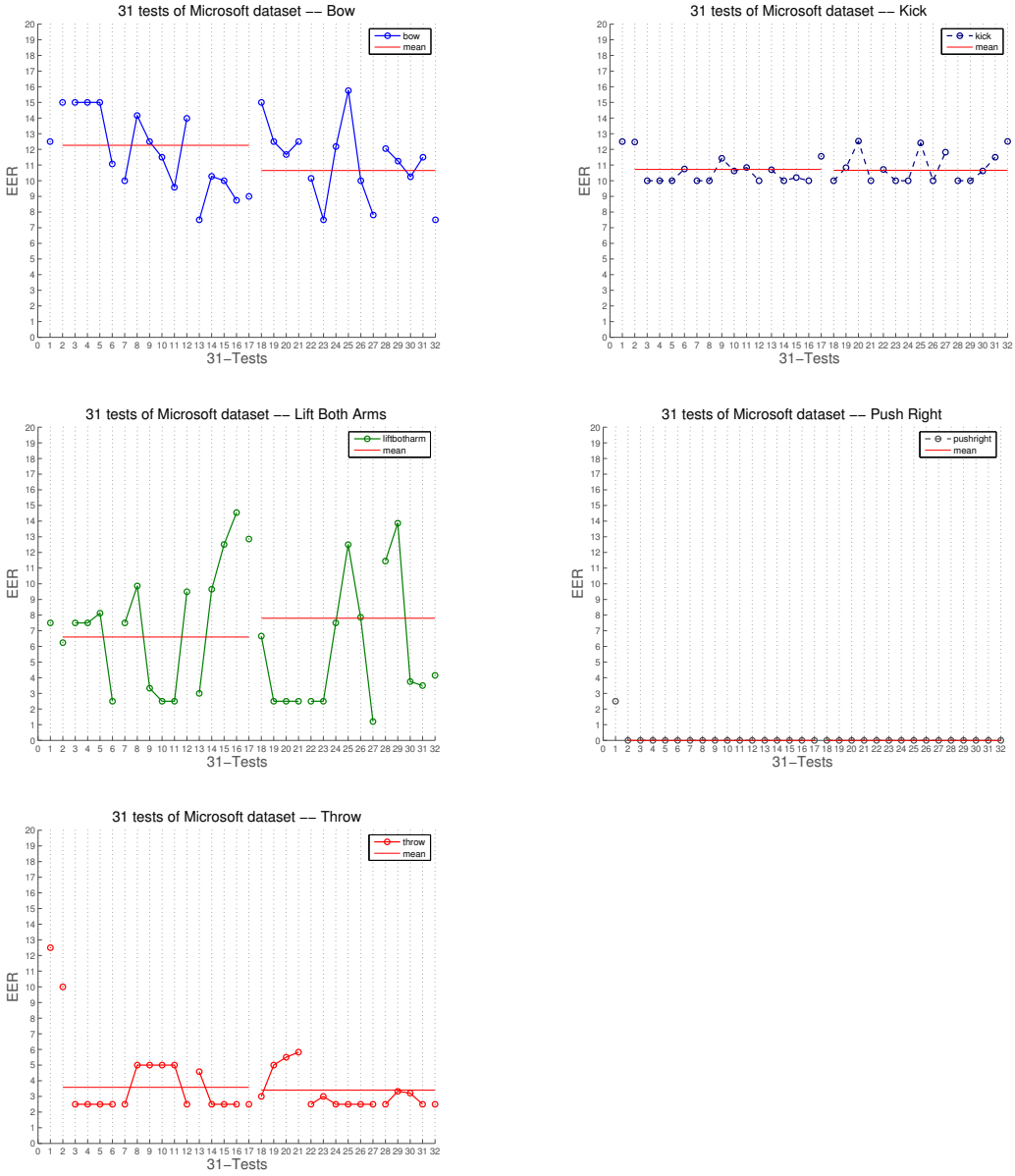

Figure 6: Results of EER for 31tests of Authentication on Microsoft dataset

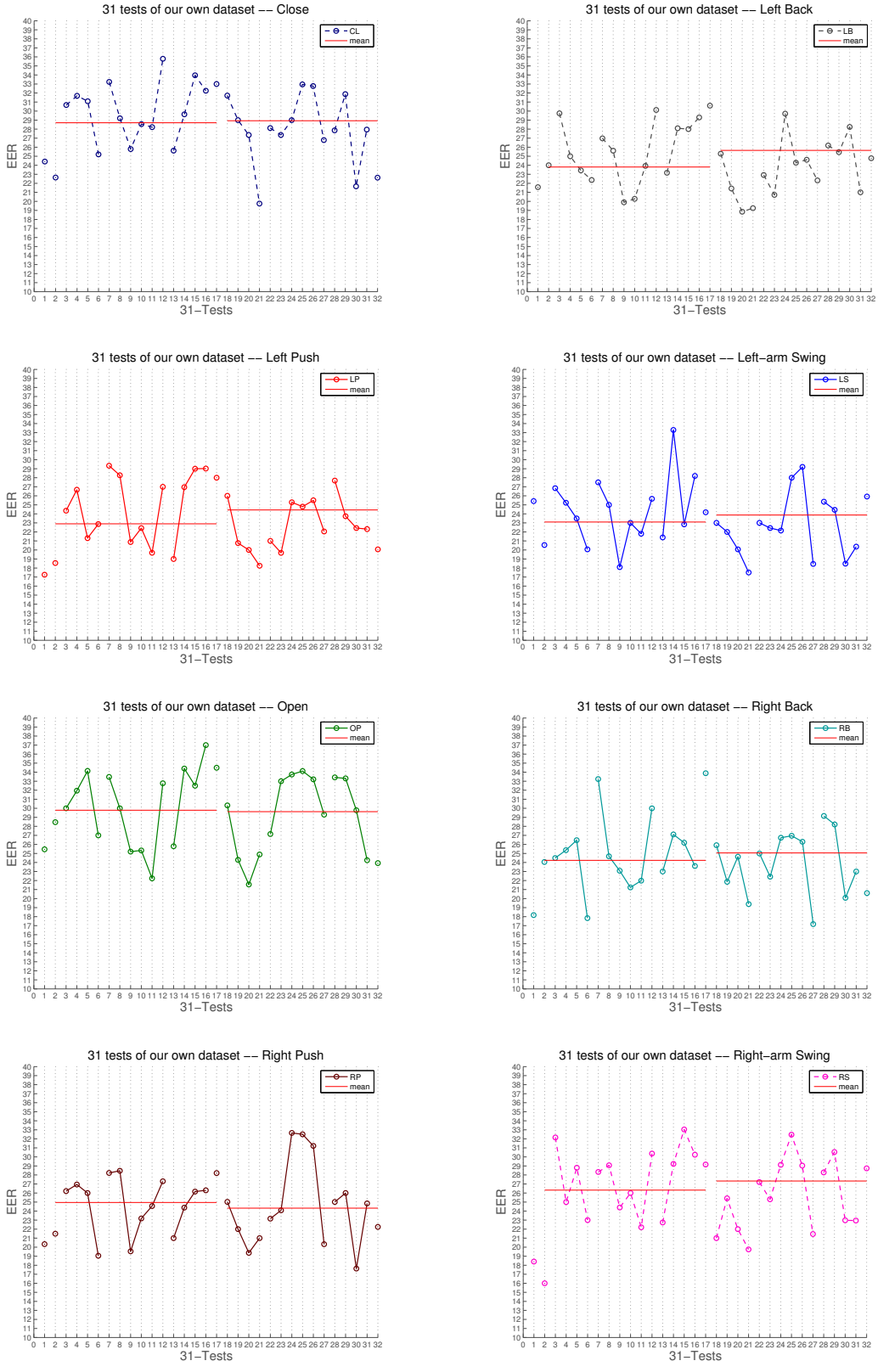

Figure 7: Results of EER for 31tests of Authentication on our own dataset

5 Result Analysis

For user identification, we can see from the tables of both baseline experiment and advanced experiment that EER decreases when number of unauthorized user increases, i.e, number of authorized user decreases.

In baseline experiment, for our own dataset, EER performance of the silhouette sequence feature are much better than skeleton sequence feature for almost 30%. This is not surprising since silhouette sequences have much more samples for features in a single frame than skeleton sequences which have only 30 samples for features in a complete gesture.

Considering the calculation method of our log-covariance matrix, if the number of sample for features is sparse, we will have negative eigenvalues with very small absolute values of the covariance matrix and thus complex log eigenvalues of the log covariance matrix. To address this issue, we "drop" those negative eigenvalues, i.e, setting those negative eigenvalues to a very small positive eigenvalues then taking logarithm of the updated eigenvalues. This solution came from the idea of PCA. In PCA, eigen decomposition is performed to get the eigenvalues and eigenvetors, we preserve the first k most biggest eigenvalues and their corresponding eigenvectors and drop the other eigenvalues, since these k eigenvalues and eigenvectors remain most of the information that contains in a matrix. However, when we drop those negative eigenvalues, we kind of lose information. In the meantime, log-eigenvalues would become very large real negative numbers, which makes the entries in the log covariance matrix very large. All these steps result in high EER performance at the end.

Compared with the results of skeleton feature of own dataset, EER of skeleton feature of Microsoft dataset are much more smaller for almost 25%. One reason is that gestures in Microsoft dataset have more samples for features than gestures in our own dataset. Each gesture in Microsoft dataset lasts for at least 60 frames, while all gestures in our own dataset last for 30 frames. Moreover, there's a big difference in the gesture style in two dataset. We can find that all the gestures in our own dataset are arm-related gestures, which means that people stand still and only move their arms. While in Microsoft dataset, all gestures are whole body gestures. This brings out the idea that a more complex gesture involving the movement of whole body is more suitable for identification system.

In advanced user identification experiment, the results are so random. The EER of some gestures are getting better when adding time reference or deleting joints, while some are getting worse. Thus, we can't make any detail conclusion from these results. But one gesture, "push right" in Microsoft dataset got best results when adding time reference and deleting joints. EER are all zero in 31 tests. It is interesting that this "push right" gesture is a arm-related gesture, but has much better

EER performance compared with all "arm-related" gestures in our own dataset. The reasons are that first, when people doing "push right" in Microsoft dataset, they also move their body rather than standing still, while in our dataset, most of people stand still while doing gestures. Second, Microsoft dataset has over 60 frames for one complete gesture, while in our own dataset, there are only 30 frames for one single gesture.

For user authentication, the EER performance won't change as number of unauthorized user increases. Here we provide the proof.The definition of FAR and FRR are given by the following equations:

$$\begin{aligned} FAR &= \frac{\#\ false\ acceptance}{\#\ total\ false\ acceptance\ tests} \\ FRR &= \frac{\#\ false\ rejection}{\#\ total\ false\ rejection\ tests} \end{aligned} \tag{7}$$

Assume we have $L/(K-L)$ split, L unauthorized users and $(K-L)$ authorized users, and threshold θ is fixed. Set r to be the number of realizations of each gesture, U_i denotes the user index in the unauthorized group and A_j denotes the user index in the authorized group, where $i = \{0, 1, 2 \cdots, L\}, j = \{0, 1, 2 \cdots, K-L\}$, $FA_{U_i}^{A_j} = \{0, 1, 2 \cdots, r\}$ denotes the number of false acceptance when comparing each realization of one gesture of one user U_i to all the realizations of one gesture of one user A_j. FA and FR tests should be done on all possible combinations. For e.g, we have 20 users in total, and we have (1/19) split, then user1, user2, $\cdots$, user20 has to be in the unauthorized group by turns. Moreover, $FA_{Useri}^{Userj} = FA_{Userj}^{Useri}$, and we denote this value as $FA_{i,j}$. Once θ is fixed, this value is fixed. Therefore, when users are divided into unauthorized and authorized groups under all combinations, $FA_{i,j}$ could be calculated multiple times.

Similarly, $FR_{A_j}^{A_j} = \{0, 1, 2 \cdots, r-1\}$ denotes the number of false rejection when comparing each realization of one gesture of the authorized user to all remaining realizations of this gesture of this user himself, where $j = \{0, 1, 2, \cdots, K-L\}$. $FR_{A_j}^{A_j}$ could be calculated multiple times.

Back to equation (7), we have:

$$\begin{aligned} FAR &= \frac{\begin{pmatrix} K-2 \\ L-1 \end{pmatrix} \times \sum_{i,j} \mathbf{1}\{FA_{U_i}^{A_j} > 0\}}{\begin{pmatrix} K \\ L \end{pmatrix} \times (K-L) \times K \times r} \\ FRR &= \frac{\begin{pmatrix} K-1 \\ L \end{pmatrix} \times \sum_{j} \mathbf{1}\{FR_{A_j}^{A_j} > 0\}}{\begin{pmatrix} K \\ L \end{pmatrix} \times (K-L) \times (r-1)} \end{aligned} \tag{8}$$

where, $\begin{pmatrix} K-2 \\ L-1 \end{pmatrix}$ is the number of times that $FA_{i,j}$ has been calculated in all the tests when θ is fixed. $\begin{pmatrix} K-1 \\ L \end{pmatrix}$ is the number of times that $FR_{j,j}$ has been calculated in all the tests when θ is fixed.As stated above, once θ is fixed, $FA_{U_i}^{A_j}$ and $FR_{A_j}^{A_j}$ won't change. Thus, $\sum_{i,j} \mathbf{1}\{FA_{U_i}^{A_j} > 0\}/r = const1$, and $\sum_{i,j} \mathbf{1}\{FR_{A_j}^{A_j} > 0\}/(r-1) = const2$. Back to equation (8), we have:

$$
\begin{aligned}
FAR &= \frac{\begin{pmatrix} K-2 \\ L-1 \end{pmatrix} \times \sum_{i,j} \mathbf{1}\{FA_{U_i}^{A_j} > 0\}}{\begin{pmatrix} K \\ L \end{pmatrix} \times (K-L) \times K \times r} = \frac{\begin{pmatrix} K-2 \\ L-1 \end{pmatrix} \times const1}{\begin{pmatrix} K \\ L \end{pmatrix} \times (K-L) \times K} \\
FRR &= \frac{\begin{pmatrix} K-1 \\ L \end{pmatrix} \times \sum_{j} \mathbf{1}\{FR_{A_j}^{A_j} > 0\}}{\begin{pmatrix} K \\ L \end{pmatrix} \times (K-L) \times (r-1)} = \frac{\begin{pmatrix} K-1 \\ L \end{pmatrix} \times const2}{\begin{pmatrix} K \\ L \end{pmatrix} \times (K-L)}
\end{aligned} \tag{9}
$$

Finally, we have:

$$
FAR = \frac{const1}{K(K-1)}, FRR = \frac{const2}{K} \tag{10}
$$

As we can see from equation(10), FAR and FRR are only related to K, which explains why FAR and FRR won't change. Thus, EER won't change as it is achieved when FAR equals to FRR. To get EER value, we plot a line $y = x$ on the figure of FAR *v.s* FRR, and manually get the intersection point (EER), which makes a little fluctuate on EER but theoretically EER won't change in user authentication experiment.

In baseline experiment, for our own dataset, EER results of silhouette sequence are smaller than results of skeleton sequence for almost 15%. Compared among results of skeleton features of both dataset, Microsoft dataset again beats our own dataset. In advanced user authentication experiment, we can see again that Microsoft dataset has EER performance 10% smaller than our own dataset. From the advanced user authentication results figures in chapter 4, we can still see that the EER results are so random no matter how we choose the $L/(K-L)$ split. The reasons are the same as described in the above advanced user identification paragraph.

From all the tables and figures, we can find a general phenomenon that EER performance of user authentication experiment are generally better than user identification experiment. The reason is that since we are using the method of leave-one-out cross validation (LOOCV), the total number of tests in user authentication are more than the total number of tests in user identifications. Therefore, back to the definition of FAR (FRR) and EER, FAR (FRR) equals to total number of false acceptance (false rejection) divided by total number of tests, with larger denominator, FAR (FRR) of user authentication tests would be smaller.

6 Conclusion

So far, we developed baseline experiment and advanced experiment of user identification and user authentication on both our own dataset and Microsoft kinect dataset. Log covariance matrix is the core algorithm used in these experiments. Two kinds of feature, silhouette sequence feature and skeleton sequence feature, are used to calculate the log covariance matrix. By calculating the log covariance matrix of each gesture, we achieve the distance of two gestures by comparing the Frobenius norm of their log covariance matrices. False acceptance (FA) and false rejection (FR) test are conducted. False acceptance rate (FAR), false rejection rate (FRR), and equal error rate (EER) are defined to evaluate the performance of experiments.

The results of user identification on our own dataset show that silhouette sequence feature is much more powerful than skeleton feature. During the eigen decomposition of covariance matrix of skeleton features, we encounter negative eigenvalue with very small absolute value, i.e complex log eigenvalue. To solve this issue, negative eigenvalues are set to a very small real positive number, for e.g 10^{-6}, then keep doing the rest process. This idea came from PCA. By doing this, we make entries of log-covariance matrix very big and also lose information. Thus, the results of skeleton sequence feature are much worse than results of silhouette sequence feature. Comparing the results between baseline experiment and advanced experiments, we are difficult to draw detail conclusion since advanced experiments have random results. It's not always that EER get better when adding time reference or deleting skeleton joints. But one gesture, "push right" in Microsoft dataset shows the best EER, i.e EER equals to zero, when adding time reference and/or deleting joints.

All the results of this project show that log covariance matrix algorithm is robust when number of samples for feature is large. While, it is less powerful when number of samples is small. Also, adding time reference has no significant effect on log covariance matrix. When considering using time reference to do authentication or identification, people should choose other methods like DTW. [3] shows the encouraging results of user authentication via DTW. It is an insight that gestures which are more arm-related and involving other body moments at the same time, like waving arms and gently shaking body meanwhile, would be powerful in future user authentication system design. Also, the length of a complete gesture should be long enough. Moreover, it would be very interesting to use multiple kinects or cameras to capture more features of gestures. 3D image reconstruction may be a good way to do user authentication. More thorough work would be developed on these topics in the future.

References

[1] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *CHI*, ACM, 2012.

[2] K. Lai, J. Konrad, and P. Ishwar, "Towards gesture-based user authentication," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, 2012.

[3] J. Wu, J. Konrad, and P. Ishwar, "Dynamic time warping for gesture-based user identification and authentication with kinect," in *Acoustics, Speech, and Signal Processing (ICASSP), 2012 IEEE 38th International Conference on*, 2013.