Gesture recognition in 3D space using dimensionally reduced set of features

Łukasz Gadomer, M. Skoczylas

Faculty of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
{l.gadomer,m.skoczylas}@pb.edu.pl
http://www.wi.pb.edu.pl

Abstract. In this study authors present a solution to track and recognize arbitrary gestures of hands in three dimensional space and review the recognition accuracy. The idea of this novel gesture recognition system is described and results of research made on a recorded gesture data set are presented. Gesture instances were defined by user standing in different distances from the controller, in different placements of their field of vision and with different speeds, making recognition velocity and position invariant. Authors' goal was to find the minimal number of features that give satisfying gesture classification result in order to achieve a compromise between accuracy and computation time. In this publication progress of the research on gesture recognition problem is described and a comparative study is presented.

Keywords: gesture recognition, features, dimensionality reduction, singular value decomposition

1 Introduction

This paper presents a solution which allows tracking hand gestures in three dimensional space that can be inserted into a CAVE3D (Automatic Virtual Environment, see Figure 1).

System consists of two main parts: a gesture recognition tool and a graphical environment. The gesture recognition tool allows user to create gestures database, learn it using one of selected classifiers and then recognize gestures performed by the user in real time. Implemented solution allows recognition of gestures recorded with varied velocity and with different user placement relating to the controller, so it is velocity and position invariant. It also contains build–in features that test recognition accuracy which are helpful during research and tests of the quality of this solution.

The whole system allows real–time position and velocity invariant gesture recognition: preparation of user's own set of gestures, classifiers learning, and then recognition of gestures in real–time using selected classifiers, it is a novel and innovative solution.

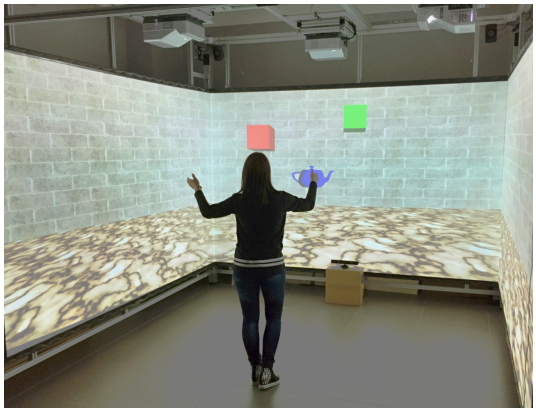

Fig. 1. Cave3D experimental setup consists of 3 screens with 3D projectors that are surrounding the user.

The main purpose of the work described in this paper is to reduce size of the data set and computation time of gesture recognition while maintaining the highest possible classification accuracy. To achieve this, number of attributes was reduced by extracting features from a prepared data set, performing dimensionality reduction and checking what is the minimal number of features needed to achieve satisfying accuracy. This goal was achieved, and the important issues concerning this work are described in the following paragraphs.

This work is the continuation of researches presented in [6] and [4].

2 Related work

The problem of gesture recognition is a challenging and popular issue. Many researchers tried to resolve it in their own way. Diversity of approaches is, inter alia, a consequence of different possible ways of gesture representation. Very often gesture is represented as a movement of single body's part, usually a hand. This approach is the same that was described in the following gesture definition: *gestures are* ,,movements of the arms and hands which are closely synchronized with the flow of speech" [3].

In [7] authors studied possibility of gesture recognition using accelerometer MEMS (Microelectromechanical System). This device was controlled by user's hand and its movement in three dimensions was observed. Authors performed their research on seven simple gestures. The same device was used in [1] to resolve similar problem. In that work, the gesture classification was realized using Dynamic Time Warping (DTW) algorithm. A database consisting of 3780 instances of gestures, grouped into 18 different decision classes was used. Another solution which is based on the accelerometer device and the same classification algorithm is described in [8]. In this publication authors tested classification accuracy in two cases: user-relevant and user-irrelevant. Their research database

contained 3200 instances, each of them represented one of eight different gestures. In this case every single instance of gesture took the same time (3,40 seconds). In all these solutions gestures were treated the same way as presented in following paragraphs – as a movement of one single point, which represents user's palm in three dimensional space.

Another example of tracking hand's movement was described in [9]. In this case to track gestures authors used a finger–worn device which was similar to a ring – it was called Magic Ring (MR). They presented personalized gesture recognition using a method of adaptive template adjustment. Another exemplary gesture recognition solution is called 1$ [5]. This name was given to symbolize algorithm's low cost and simplicity – it's implementation took only about a hundred lines of code. According to authors' description it works satisfyingly even there is only one training instance of each gesture. This work was an inspiration for the authors of [2]. They designed a solution based on the algorithm described in [5] which used Sparse Representation (SR) and Compressed Sensing (CS) methods.

3 Gesture data collecting and processing

In this section authors describe issues connected with gesture data collection and processing. That was described in details in [6] and [4], however let's outline briefly here: first, user has to prepare his own data set. He stands before the controller and performs gestures, signaling beginning and end of the gesture[1] Then he can save created data set and use it for two purposes:

1. Learn selected classifier and use it to perform gesture recognition in real time,
2. Use selected data set for classifier's parameters optimization and measure the recognition accuracy to evaluate solution quality.

The first issue was described widely in [6]. In this publication authors would like to concentrate on the progress which concerns the second one.

The issues connected with position and speed invariance was widely described in [6] and [4]. For this reason, we do not concentrate about them in this paper.

3.1 Gesture dataset

Gesture database included 12 different gestures shown in Table 1, recorded as values in relative data format. Each gesture type (a decision class) was recorded 80 times which in total sums up to a gesture database consisting of 960 gestures. All of gestures were performed by four different users – each user recorded 20 gestures. What is more, they were asked to perform gestures in a different

[1] It can be done by clicking ,,start recognition" and ,,end recognition" in our software. It can be done, for example, with a help of the operator, who can decide when are start and finish moments and that method was used in the data collecting process.

way and change their positions a bit between every gesture. As a result, every recorded instance was a bit different than others. Gestures were recorded with different velocities, users were standing in different distances to the controller and they were placed in a different parts of detecting range. Such way of performing gestures provided research data that allows test of position and velocity invariance in a real scenario. In addition, recordings that were not perfect were also included into the dataset (but recordings were not repeated), making the data set even more difficult to analyze. The only limitation in recording instances was assumption that every gesture should be performed in the same direction. It means that, for instance, every horizontal line is performed from left side to right side.

As it is shown in Table 1, all gestures are two–dimensional by their defintion (for example, brackets are designed to be two–dimensional, etc.). However they are captured in three–dimensional space. Device captured depth, that was recorded and written to the dataset the same way as width and height. It means that it was also important if the data collecting participants were performing movements in depth dimension. We have chosen such gestures, but there would be no problem to choose, for examples, "push" and "pull" ones – whichever selected gestures would be, the algorithm should work the same way.

All of recorded gestures were shuffled and written into a single dataset. The information about gesture performer was not saved. It means that gesture recognition is fully user–independent.

Gesture shape	Starting point
(	Top
)	Top
<	Top
>	Top
∧	Left
\	Top
/	Top
|	Top
—	Right
∼	Left
O	Top
8	Top

Table 1. Gestures dataset

3.2 Feature data representation

Number of attributes in a data set depends on the length of gesture. Authors assumed that all gestures have 40 samples, three dimensions each, which makes 120 attributes. Recording 40 samples using Kinect controller takes a bit more

than one second. It does not mean that user has to perform the gesture in exactly one second – it can take any amount of time. This assumed number is just a final length of the gesture after scaling.

To resolve this problem authors decided to extract features from the gesture instances. The advantage of this solution is the fact that it is possible to extract a given number of features independently from the gesture length. It means, no matter how long the gesture is, the number of features is always the same. This allows reduction or extension of dimension to the same number of dimensions, allowing to have unlimited length of gestures.

To extract data features authors decided to transform the prepared data set following way: from a relative hand position absolute values were computed, but always starting from point (0, 0, 0). That means gesture's samples values were translated to the beginning of the coordinate system. Authors performed this operation to express real movements of the hand – representation used in [6] was a proper one for direct recognition, but in authors' opinion it needs above transformation to achieve features that express the given problem best way.

Table 2 shows features extracted from the prepared dataset. Popular statistical and signal features were selected. As it is presented, most of these features were computed independently for each axis and for all of the axes together. Axis to axis features were computed between the cartesian of axes. In total 49 features were extracted. In the Table 2 n is the number of samples k and l are the sample pair of axes and a is the sample.

4 Dimensionality reduction

One of the main objective in this publication is to check whether the minimal number of features exists that allows to achieve rewarding gesture classification accuracy. As it was mentioned in section 3.2, 49 features were extracted. This is the maximal number of dimensions proposed in our computations. The next step is the dimensionality reduction, which objective is to reduce number of features. To achieve this a Singular Value Decomposition (SVD) algorithm was used, but considering only real numbers (which is a right assumption for the purposes of our gesture recognition problems, where numbers cannot be complex).

The singular value decomposition of $m \times n$ matrix M is a $M = U\Sigma V^*$ factorization, where:

- U is a $m \times m$ unitary matrix,
- Σ is a $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal,
- V^* is a $n \times n$ unitary matrix, which is a transposition of V.

Values that are placed on the diagonal of Σ are called singular values of matrix M. The m columns of U are known as left–singular vectors of M and the n columns of V are called right–singular vectors of M.

First, the SVD algorithm is performed on all set of n features. Then, to reduce this data set to k dimensions, all elements of $k + 1$ to n columns of Σ

Feature	Description	Equation	Computed for
Average	Sum of samples divided by number of samples	$\frac{1}{n}\sum_{i=1}^{n} a_i$	Each axis, all of the axes
Standard deviation	Measure that is used to quantify amount of dispersion of a set of samples	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i-\bar{a})^2}$	Each axis, all of the axes
Variance	Measure that expresses how far a set of samples is spread out	$\frac{1}{n}\sum_{i=1}^{n}(a_i-\bar{a})^2$	Each axis, all of the axes
Ratio	Relationship between range of two sets of samples	$\frac{\max(k_i)-\min(k_i)}{\max(l_i)-\min(l_i)}$	Each pair of axes
Covariance	Measure that expresses how much two sets of samples are related	$\frac{1}{n}\sum_{i=1}^{n} kl - \bar{k}\bar{l}$	Each pair of axes
Correlation	Measure that expresses how much two sets of samples are related	$\frac{\sum_{i=1}^{n} kl - \bar{k}\bar{l}}{\sqrt{\sum_{i=1}^{n}(k_i-\bar{k})^2 \sum_{i=1}^{n}(l_i-\bar{l})^2}}$	Each pair of axes
Skewness	Measure of asymmetry of set of samples	$\frac{\sqrt{n}\sum_{i=1}^{n}(a_i-\bar{a})^3}{\sqrt{\sum_{i=1}^{n}(a_i-\bar{a})^2}^3}$	Each axis, all of the axes
Kurtosis	Measure of tailedness of set of samples	$\frac{n\sum_{i=1}^{n}(a_i-\bar{a})^4}{\sqrt{\sum_{i=1}^{n}(a_i-\bar{a})^2}^4}$	Each axis, all of the axes
Signal magnitude area	Measure of magnitude of set of samples	$\sum_{i=1}^{n} x_i$	Each axis, all of the axes
Signal magnitude vector	Measure of degree of movement intensity	$\sum_{i=1}^{n}\sqrt{x_i^2+y_i^2+z_i^2}$	All of the axes
Root mean square	Measure defined as a square root of mean of squares of a sample	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}$	Each axis, all of the axes
Mean deviation	Average of absolute deviations from a central point of set of samples	$\frac{1}{n}\sum_{i=1}^{n}\lvert a_i-\bar{a}_i\rvert$	Each axis, all of the axes
Interquartile range	Difference between the upper and lower quartiles of set of samples	$Q_3 - Q_1$	Each axis
Energy	Energy of set of samples	$\sum_{i=1}^{n}\lvert a_i\rvert^2$	Each axis, all of the axes

Table 2. Extracted features

and V matrices and rows of U matrices are set to zeroes. Then a new M' matrix is computed according to the procedure presented before. As a result, its first k columns are different from 0 — these columns form a new, dimensionally reduced set of features.

5 Research description

For accuracy testing purposes and to achieve best results in gestures recognition problem, authors performed several experiments with collected gesture data. Aim of this study was to check the classifiers, primarily to identify accuracy in different gestures recognition, as well as speed of calculations.

5.1 Parameters optimization

The basic issue connected with classification using many classifiers, expecially SVM, is a parameter optimization. This process is essential due to the fact that

classification accuracy highly depends on parameters of the classifier. Parameters have to fit the character of data. It is a serious problem because there is no simple way of selecting proper parameters. A popular method to obtain kernel parameters is a grid search. Note, that it is also possible that selected parameters do not fit to the testing set. All these facts mean that parameter optimization does not have a perfect solution – choosing good parameters is rather a compromise than a sure answer.

To minimize risk of data fitting authors decided to perform parameter optimization, use a single random data set division (but the same each time) and the 5–fold cross-validation. This division assumes that in every one of the five parts there are the same number of each gesture class instances. For each parameter combination the dataset was randomly divided into five parts, but taking into account that described assumption. Classifier is then learned using four of these parts and tested using the fifth, out–of–bag (OOB) part. This process is repeated five times (each time the other part is OOB part) and then the result is averaged. The same actions are performed for each of parameters combination and the best one is selected.

Because of long time of computations, the parameter optimization procedure was performed in a parallel way. The parallelisation ratio was computed on a single personal computer with Intel i7 processor (8 cores, 16 threads).

5.2 Classification

After selection of best classifiers' parameters (and kernel function parameters for the SVM classifier) authors performed data classification using the obtained parameters set. Similarly to parameter optimization, 5–fold crossvalidation was used, but the classification with 100 different random divisions was performed, not only the single one. The whole classification process was the same that it was in the case of parameters optimization – the difference is that research results were averaged over all these 100 divisions, and that value was recorded as a final classification accuracy.

5.3 Research parts – two experiments

Authors performed two main parts of the research and recognition accuracy evaluation.

The first experiment concerns analysis how tested classifiers deal with gesture classification problem. Evaluation of all four classifiers in two cases was performed: with and without the normalization. Additionally, for SVM classifier, five kernel functions were tested independently. Such research gave a large number of results, and they are summarized and presented in section 6.

For the latter, a Singular Value Decomposition was used to find the minimal number of features that give satisfying gesture classification results in order to obtain a compromise between the accuracy and computation time. To achieve this, gesture classification accuracy with the increase of number of dimensions was compared. In addition, best results using full data representation to feature

data representation were compared, in order to check if new way of expressing data does not cause the severe drop of the classification accuracy.

One of the main purpose of second experiment was to check how many features are enough to achieve satisfactory classification accuracy. To accomplish this, dimensionality of data set was reduced, so that each example with 49 features was reduced iteratively into 48 new data sets that consisted from 1 to 48 features. Each of these data sets was tested using method described above. This allowed us to judge how an addition of a single dimension to a data set affects the classification accuracy.

All the results obtained are presented and discussed in section 6.

6 Results and discussion

6.1 First experiment

First experiment was performed using relative data representation. For each classifier one configuration was selected that achieved best results based on recognition accuracy comparison, and then these were compared with other classifiers' best configurations. Summarized results are presented in Table 3.

Classifier	SVM	NN	RBF	LMT
Average classification accuracy	95.85	92.74	92.16	90.04
Mean standard deviation of the accuracy	3.74	5.60	6.00	6.11
Calculations Time	207.41	21819.19	201.23	5630.90

Table 3. Results of measurements obtained using selected classifiers

On the basis of these observations, authors conclude that for the given problem of classification of gestures the best results are obtained using the SVM classifier. SVM performed best in the shortest possible time and was characterized by a low diversity of the results achieved in subsequent repetitions.

The results of SVM kernels comparison are shown in Table 4. As we can see, in all the cases accuracy obtained using normalization is better than without using it. The best result was produced using wavelet kernel, but the difference between this kernel and the others was not large. It is important to note that without normalization Wavelet kernel generally gives much worse results. Authors have chosen best possible parameters and final results were good, but without cross validation it would be really hard to choose because most of parameters without normalization yielded bad results with the Wavelet kernel.

Considering results of presented research it is also worth to note what are the reasons of recognition mistakes. Figure 2 shows classification errors for opening bracket gesture. The most problematic were gestures similar to less–than sign and often they were incorrectly recognized as a vertical line. It is vital to note visual similarity between these gestures. When the user marks the curve too

Kernel	Lienar	Polynomial	Radial	Sigmoid	Wavelet
Without normalization	94.30	94.46	95.50	95.00	95.62
With normalization	95.01	95.36	95.46	95.69	95.85

Table 4. Results of measurements obtained using selected SVM kernels

sharply while performing opening bracket gesture, it makes similar to less–than sign. When user marks this curve not sharply enough, gesture starts to look like a vertical line gesture. This explains reasons of classification errors. Analysis of incorrectly classified instances of other gestures confirmed that observation.

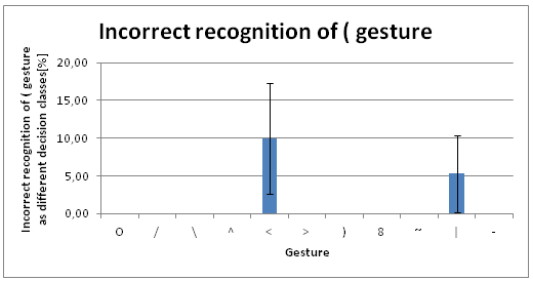

Fig. 2. Incorrect recognition of opening bracket gesture as different decision classes

6.2 Second experiment

The first tested approach was a check difference between classification accuracy using relative data representation, achieved in the first experiment, and the proposed feature representation using 49 or less proposed features. This was checked for each of five proposed kernel functions. The results are presented in Table 5.

Kernel Function	Relative data [%]	Features [%]
Linear	95.01	94.15
Polynomial	95.36	94.15
Sigmoid	95.46	77.90
RBF	95.69	92.85
Wavelet	95.85	93.11

Table 5. Comparison of relative data representation and feature data representation classification accuracy

As it is shown in section 5, for four of five kernels the difference was about 1.5%–3% (it was bigger for sigmoid kernel). It is a noticeable drop of classification

accuracy, which confirms that feature extraction causes loss of some information. The other reason can be not perfect choice of features that were extracted – this can be checked in further research. On the other hand, by performing feature extraction we reduced the number of dimensions more than twice (from 120 to 49), as a result we also reduced the classification time. We judge the 1.5%–3% difference is a price worth to pay for more than twice reduction of computation time.

The main part of our research dealt with classification accuracy using data sets that consisted of different number of dimensions. Authors checked 49 data sets having number of dimensions from 1 to 49 (with a step of 1). The results are presented in Figure 3.

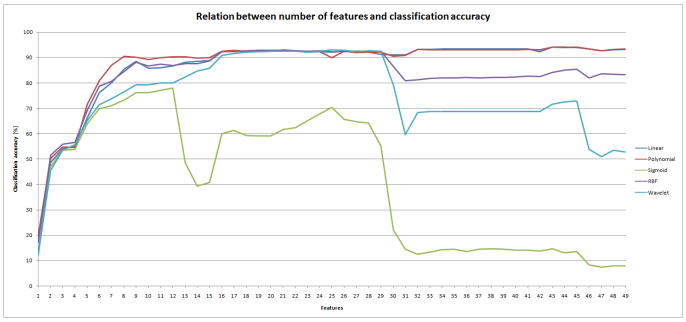

Fig. 3. Classification accuracy referring to the number of dimensions

First of all, addition of each dimension is significantly increasing the classification accuracy for each kernel, but this tendency stops after 7–12 dimensions. At this point classification gains stable and satisfactory results. Best results most kernels (instead of sigmoid) started to achieve at 16th dimension. The further increase of dimensions from 16 to 31 does not result in significant classification accuracy growth, which means next features do not provide any more important information about data. After 31th dimension in three of five kernels classification accuracy drops, that means some information is excessive and is bringing unnecessary noise to the data set for these kernel functions.

The best results in 16–31 dimensions were comparable for each kernel function, but not sigmoid. Slightly better in this range is a wavelet function. In the larger number of dimensions the best classification accuracy was achieved by linear and polynomial kernel functions, and they achieved best results in the whole research. Sigmoid functions, comparing to the other ones, gave unsatisfactional results. We were unable to select a correct set of parameters for this function to achieve results comparable to other ones.

In Figure 4 the best results achieved during the parameters optimization process are shown. Figure 5 shows differences between classification accuracy achieved during optimization of parameters.

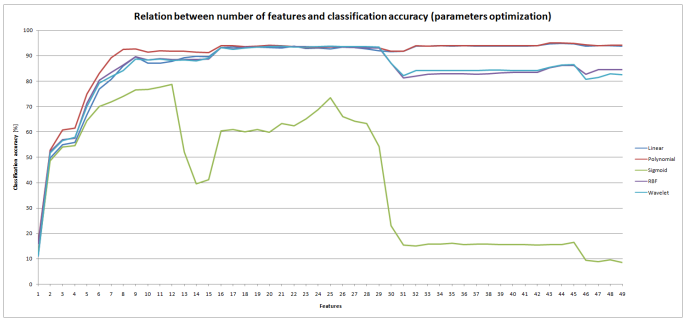

Fig. 4. Classification accuracy referring to the number of dimensions — parameter optimization

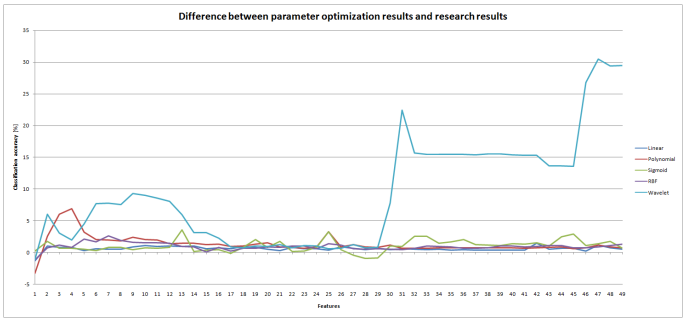

Fig. 5. Differences between parameter optimization accuracy and research accuracy

In almost all of the cases results achieved using parameters optimization were better than during the research. Differences are the result of data overfitting. For four of five kernels (instead of the wavelet one) the differences were oscilating about 0%–3% all the time. For wavelet kernel the differences were much larger. For 0 to 30 dimensions they did not exceed 10%. For the larger number of dimensions (above 30) it was oscilating between 15% and 30%. It means this kernel function is the most sensitive to selection of parameters.

7 Conclusion

The method and algorithm of real-time gestures recognition described in this paper can be inserted into the CAVE3D system. Gestures can be successfully recognized using classifiers. Selection of appropriate classifier to solve the problem of gestures recognition is crucial. Based on studies presented in this paper it can be concluded that the decision should fall on the SVM classifier. It should be emphasized however, that results could be slightly different for different sets of gestures or other selected classifiers parameters, but taking into account specific

nature of the problem and carefully conducted study by authors, the result of them can be considered as representative for a given research problem.

Also, according to the research presented in this paper, only 16 features are enough to achieve results that are about 1.5%–3% worse than using full data representation. This means that it is possible to reduce data set size about 7–8 times for slightly lower and probably unnoticeable cost of the classification accuracy.

Authors tested selected classifiers and found the best one that fits gesture recognition problem. Then, using this classifier, authors proved that it is possible to reduce the number of data set dimension using different feature data representation. The minimal number of features which gives satisfying result was also found for the data set used in this research.

Acknowledgment

This work was supported by the grant S/WI/1/2013 from Bialystok University of Technology founded by Ministry of Science and Higher Education.

References

1. A. Akl, C. Feng, and S. Valaee. A novel accelerometer-based gesture recognition system. *Signal Processing, IEEE Transactions on*, 59(12):6197–6205, Dec 2011.
2. A. Boyali, and M. Kavakli. A robust gesture recognition algorithm based on sparse representation, random projections and compressed sensing. In *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, pages 243–249, July 2012.
3. D. McNeill. *Gesture and Thought*. University of Chichago Press, Chicago, USA, 2007.
4. Ł. Gadomer. Towards gesture recognition in three-dimensional space. *Advances in Computer Science Research*, Nr 12:5–20, 2015.
5. J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *ACM Symposium on User Interface Software and Technology (UIST '07). Newport, Rhode Island*, pages 159–168, July 2007.
6. Ł. Gadomer, M. Skoczylas. Real time gesture recognition using selected classifiers. *Architecturae et Atribus*, 6(1):14–18, 2014.
7. R. Xu, S. Zhou, and W. J. Li. Mems accelerometer based nonspecific-user hand gesture recognition. *Sensors Journal, IEEE*, 12(5):1166–1173, May 2012.
8. S. M. A Hussain, and A. B. M. H. Rashid. User independent hand gesture recognition by accelerated dtw. In *Informatics, Electronics Vision (ICIEV), 2012 International Conference on*, pages 1033–1037, May 2012.
9. Y. Zhou, D. Saito, and L. Jing. Adaptive template adjustment for personalized gesture recognition based on a finger-worn device. In *Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on*, pages 610–614, Nov 2013.