User-Independent Real-Time Hand Gesture Recognition Based on Surface Electromyography

Frederic Kerber
German Research Center for Artificial Intelligence (DFKI)
Saarland Informatics Campus
Saarbrücken, Germany
frederic.kerber@dfki.de

Michael Puhl
Saarland University
Saarland Informatics Campus
Saarbrücken, Germany
s9mlpuhl@stud.uni-saarland.de

Antonio Krüger
German Research Center for Artificial Intelligence (DFKI)
Saarland Informatics Campus
Saarbrücken, Germany
krueger@dfki.de

ABSTRACT
In this paper, we present a novel real-time hand gesture recognition system based on surface electromyography. We employ a user-independent approach based on a support vector machine utilizing ten features extracted from the raw electromyographic data obtained from the Myo armband by Thalmic Labs. Through an improved synchronization approach, we simplified the application process of the sensing armband. We report the results of a user study with 14 participants using an extended set consisting of 40 gestures. Considering the set of five hand gestures currently supported off-the-shelf by the Myo armband, we outperform their approach with an overall accuracy of 95% compared to 68% with the original algorithm on the same dataset.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Electromyography; gestural input; hand gestures.

INTRODUCTION
Latest advancements in mobile and especially wearable computing bring up a new generation of devices following the idea of ubiquitous computing and calm technology [19]. The smaller and smaller device sizes impose new challenges as e.g. touch input is not suitable [17]. Although speech input is on the rise[1], there are situations in which it is not appropriate, e.g. in noisy environments or when privacy matters. On the other hand, people are used to hand gestures in everyday life, e.g. due to non-verbal gestures [9]. Hence, making them usable for human-computer interaction seems a promising way. To detect such hand gestures, there exist several approaches, such as glove-based ones (e.g. [20]) or those based on computer vision [11], and combinations of these two techniques have also been investigated [18]. While vision-based systems rely heavily on environmental conditions such as lighting, glove-based approaches are often expensive and potentially limit the user's freedom of movement. As alternative solutions, approaches using bio-acoustic sensors (e.g. [4]) or surface electromyography, a technique to record and interpret electrical skeletal muscle activity non-intrusively, have been proposed. With the Myo armband[2] by Thalmic Labs, an off-the-shelf commercial device is provided that is designed to be used by people without any specific technological or medical background. Through its Bluetooth connection, no cumbersome cables are required, which further eases its application. As the armband supports only five gestures by default, we created a recording tool (see Figure 1) to gather data for a total of 40 gestures. Based on parts of the recorded data, we trained a user-independent support vector machine with a total of ten features we extracted from the raw electromyography data stream after some preprocessing steps. We further attempted to increase the recognition robustness by improving the synchronization process that is used to handle differences in position and orientation when putting on the armband.

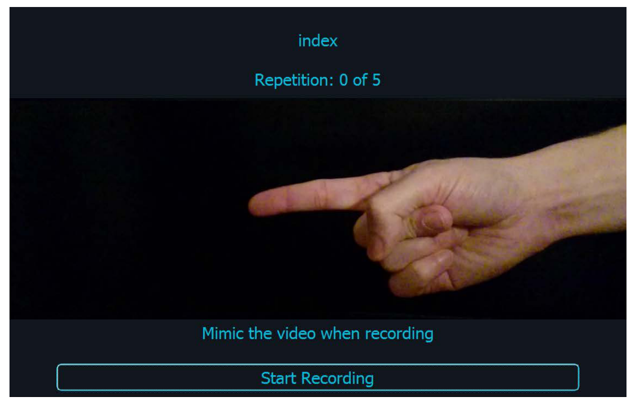


Figure 1. Recording environment used to capture the 40 gestures during our user study.

[1] http://on.mktw.net/1ZWKOBb, last retrieved 24/05/2017

[2] http://www.myo.com, last retrieved 24/05/2017

MobileHCI '17, September 04–07, 2017, Vienna, Austria

ACM ISBN 978-1-4503-5075-4/17/09...$15.00
http://dx.doi.org/10.1145/3098279.3098553

RELATED WORK

As mentioned before, glove-based approaches for hand gesture recognition such as the one by Weissmann and Salomon [20] are able to provide good recognition results of up to a 100% recognition rate, but the need to wear a glove can restrict the user's freedom of movement and might be an impediment to using the technique. In contrast, approaches using bio-acoustic sensors typically do not require the user to wear something on the hand; a wrist- or armband is often sufficient here. In [6], Laput et al. even used a standard smartwatch's accelerometer as a sensor. However, this was only possible by modifying the underlying Linux kernel, resulting in a constant CPU load for the data collection. Zhang and Harrison present an approach based on an electrical impedance tomography enabled wrist-/armband [23], providing accuracies up to 97% in a user-dependent classification. Regarding limitations of their approach, the authors mention typical issues such as environmental and sensor placement influences, including the limited applicability in a cross-user scenario.

For our work, we decided to focus on a technique that does not restrict the user's movements, is largely independent of environmental influences and provides the chance for a user-independent recognition. To ensure a wider applicability of our findings, we focused on already available hardware instead of providing a prototype device. Based on these requirements, we selected the electromyography armband by Thalmic Labs that is readily available for purchase. The topic of gesture recognition based on surface electromyography has already been examined by several independent researchers in recent years. In 2008, Saponas et al. published their first work focusing on muscle-computer interfaces [13]. Based on an off-the-shelf ten-sensor wired EMG device, they were able to distinguish finger presses w.r.t. position and pressure as well as to classify tapping and lifting gestures on one hand. In a ten-fold cross-validation, accuracies between 78% and 95% for a four-gesture set could be achieved based on user-dependent training, while as could be expected the user-independent classification performed much worse. In a follow-up work from 2009, Saponas et al. focused on classification of finger gestures for situations in which the hand is not constrained to a surface [14]. Instead of relying on a retrospective classification, a real-time approach was put in place, resulting in accuracies between 79% and 88% for the classification of four finger gestures based on user-dependent training. In their latest addition to this topic [15], the authors focused on a wireless sensing device, thereby reducing the sample rate to 5 Hz, and cross-session classification without additional training. Based on user-dependent training, accuracies up to 87% could be achieved for distinguishing three in-air pinch gestures, while classifying finger presses on a surface resulted in an accuracy of only 66% in the cross-session condition. Summarizing the three approaches from Saponas et al., there are a number of similarities to our work, but also important differences: We also focus on in-air finger gestures that do not require a surface to tap on, and utilize a support vector machine for real-time classification. While we also use an off-the-shelf wireless EMG device, today's technology enables a much faster sensor sampling rate of 200 Hz. Most importantly, we target a user-independent classification that furthermore provides the possibility to classify the gestures independent of the arm on which the armband is worn and how it is oriented. Due to these features, no more training and no exact placement of the sensing device is needed for new users. Additionally, we increase the set of considered hand gestures while preserving an accuracy of 85% to 95%, depending on the gesture set.

Based on a prototype of the device we also used in our study, Samadani and Kulić generated a user-independent hand gesture recognizer using Hidden Markov Models [12]. For ten gestures, an offline classification accuracy of 79% could be achieved. Unfortunately, the authors did not give any further details about required sensor placement accuracy.

Instead of only using eight sensors, Amma et al. investigated finger gesture recognition with an electrode array consisting of 192 EMG sensors [1]. Based on a naive Bayes classifier, the authors could show that increasing the numbers of sensors up to around 100 significantly improves the recognition. For a set of five gestures, an accuracy of 94.6% could be achieved. Furthermore, two algorithms to compensate for inter-session sensor displacements were investigated and could improve the recognition rate from 58.9% to 75%. As the algorithms rely on fixed timing information, they cannot easily be adapted for a real-time classification as we plan to implement it.

To further increase the recognition accuracy, several approaches have been investigated that combine electromyography with additional sensors, e.g. by integrating an accelerometer [7, 21, 22], an inertial measurement unit [2] or pressure sensors [8]. Although these approaches provide promising results, they bear the risk of complicating the device itself or its setup procedure or could even restrict the users in their freedom of movement. We therefore decided to base our approach solely on electromyography sensors for now.

CONCEPT & IMPLEMENTATION

We aim at providing a user-independent implementation to enable new users to directly use the system without a prior training phase. As we target a hand gesture recognition that can be used in an interactive way, e.g. to control an application or a device, we focus on fast reaction times, i.e. considerably less than one second, which we refer to as "real-time". We also focus on an underlying model that can equally be used for left- and right-handed interaction. Furthermore, we want to ensure that our system is robust against slightly different positioning and rotation of the sensing device to increase the user's flexibility when wearing the device. We extend the set of gestures that are supported off-the-shelf by the Myo armband (wave hand in/out, spread fingers, make a fist and double tap with thumb and middle finger) with six additional groups of gestures:

- Single stretched finger (5 gestures)
- Single finger is folded, the others are stretched (5 gestures)
- Tap with the thumb at a certain phalanx of another finger (12 gestures)
- Multiple stretched fingers (3 gestures)

- Flick with the thumb and another finger (4 gestures)
- More complex gestures, e.g. pinching in mid-air (6 gestures)

In total, 40 gestures were examined (see Figure 3 at the end of the paper). To collect training data for our classifier, we developed a lightweight Python application that on the one hand connects to the Myo armband using the myo-python bindings[3] to gather the raw sensor data and on the other hand, provides a graphical user interface that illustrates the gesture to mimic with a short video sequence (see Figure 1). From the software engineering point of view, we made sure that our system is easily extensible, e.g. to be able to include further gestures if technical development allows even more fine-grained distinctions. Equivalent to the original approach of Thalmic Labs, the gesture recognition is not implemented on the sensing device itself, but on a separate device the raw data is sent to.

Preprocessing

The Myo armband consists of eight EMG sensors, each providing integer values in the range from -128 to 127 with a frequency of 200 Hz. Before providing the data through the SDK, it is already filtered by a 50 Hz and 60 Hz filter to remove noise caused by power line interference. Still, the raw data contains a lot of variability caused by factors such as the amount of force used or the duration of the execution, but also aspects such as subcutaneous fat, the muscle fiber composition or the amount of hair [12] also influence the sensor readings.

To counteract effects caused by muscle contraction made as a counteraction to earth's gravitational pull, we use each channel's mean values recorded during the rest gesture. As the rest gesture is both user- and situation-dependent, this noise reduction comes at the cost of recording the rest gesture prior to actually using the device.

As we do not restrict the users to wearing the sensing device on a specific arm or with a precise alignment, we have to compensate for the resulting influences on the sensor readings. The synchronization gesture implemented by Thalmic Labs, i.e. waving the hand to the outside, provides a good estimate on the orientation of the device, as it causes a distinct high sensor value in one channel compared to all other channels. To handle left- and right-hand usage equally, a transformation of the sensor array by horizontal mirroring could be used to convert sensor readings from one side to the other.

To overcome further variability in the EMG readings, especially in cross-user comparison, we employ a normalization w.r.t. the highest measured peak. During the training phase, we consider the absolute maximum value of all channels as the highest peak. In the real-time classification mode, we consider the maximum value during the sychronization gesture as reference. In both cases, the following equation is used to adjust a measurement x_i of channel i w.r.t. to the maximum possible value (*max*) and the maximum measured value.

$$x_i' = \frac{max}{x_{max}} \cdot x_i \tag{1}$$

[3] https://github.com/NiklasRosenstein/myo-python, last retrieved 24/05/2017

To split the continuous sensor data stream into active segments, i.e. those that potentially contain a gesture, and segments without meaningful activity, we adopt an approach as presented in [22]: We compute the average absolute value of all channels x_{avg} for each sample according to Equation 2. Thereby, c is the number of channels and $x_i(t)$ denotes the measured value of sensor i at index t. Afterwards, a moving average algorithm with a configurable window size W is applied. Starting at index t, the average value over x_{avg} is calculated with respect to Equation 3. If this value exceeds a specified threshold, index t is used as the starting point of an active segment. The segment ends if all average values $x_{avg}(t)$ in a window are lower than an off threshold. These thresholds should be chosen with respect to the level of noise during a rest gesture. If the active segment is smaller then or equal to the size of a window, the corresponding segment is dropped as measurement noise.

$$x_{avg}(t) = \frac{1}{c} \sum_{i=1}^{c} |x_i(t)| \tag{2}$$

$$E_{avg}(t) = \frac{1}{W} \sum_{i=t}^{t+W-1} x_{avg}(i) \tag{3}$$

As soon as the start of an active segment is determined, we split the segment into overlapping windows as it has already been done by others [7, 13, 21]. Through this approach, we produce a time-independent signal as each window has the same length (i.e. 300 ms in our implementation). Also, the subsequent recognition steps become independent from the length of the gesture, i.e. it does not matter if a user is holding a gesture for a second or over a minute. As we treat each window independently and a gesture can be determined for each of the windows, real-time classification becomes feasible. However, the windowing of a signal can introduce unwanted aliasing effects, also called spectral leakage. To address this issue, we use the Hann function [3] to smooth the data.

Feature Extraction and Classifier Training

For the classification process, we have to extract features from the created windows that can serve as an input vector for our support vector machine. As we aim for a real-time classification, only features with low computational cost can be considered. Based on previous work (e.g. [10, 16]) we decided for ten equally-weighted features from both the time and the frequency domain (through a Fast Fourier Transformation).

Features in the time domain

For a given window of size W containing raw data x consisting of the measured values of each channel c, the following features are computed:

Root Mean Square

The Root Mean Square directly reflects the amount of muscle activity. In addition to the values for the individual channels (Equation 4), we also consider the averaged value over all channels.

$$RMS(c) = \sqrt{\frac{1}{W} \sum_{i=1}^{W} x_c(i)^2} \tag{4}$$

Mean Absolute Value

The Mean Absolute Value is computed individually for each channel (Equation 5) and then averaged over all channels.

$$MAV(c) = \frac{1}{W}\sum_{i=1}^{W}|x_c(i)| \tag{5}$$

Energy Ratio

The Energy Ratio normalizes a channel's energy E (Equation 6a) w.r.t. the first channel. The feature is computed once for each combination of two channels (Equation 6b).

$$E(c) = \sum_{i=1}^{W} x_c(i)^2 \tag{6a}$$

$$RE(i,j) = \frac{E_i/E_j}{E_j/E_1}; j = 1,...,M; i = j+1,...,M \tag{6b}$$

Histogram

The range of a channel's absolute value is divided into four equally-sized segments; the number of samples belonging to each segment is computed, resulting in four values per channel.

Variance

As the mean of the EMG signal is close to zero, the variance can be retrieved according to Equation 7.

$$VAR(c) = \frac{1}{W-1}\sum_{i=1}^{W}|x_c(i)| \tag{7}$$

Willison Amplitude

The Willison Amplitude counts the number of times the difference between two subsequent raw values exceeds a certain threshold (Equation 8). We empirically set this threshold to 30% of the maximum EMG value.

$$\begin{aligned} WAMP(c) &= \sum_{i=1}^{W-1} f(|x_c(i) - x_c(i+1)|) \\ f(x) &= \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

Zero Crossings

The Zero Crossings feature counts the number of times two subsequent raw values have different signs (Equation 9). To be robust against background noise, we only consider such crossings if the difference exceeds a certain threshold (equal to the start point detection threshold).

$$\begin{aligned} ZC(c) &= \sum_{i=1}^{W-1} f(x_c(i), x_c(i+1)) \\ f(x,y) &= \begin{cases} 1, & \text{if } x \cdot y \leq \text{threshold} \wedge |x-y| \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

Features in the frequency domain

For the frequency domain, we compute the fast Fourier transformation *fft* of the window data for each channel c. As a result, we receive the corresponding values of each frequency bin j. The total number of frequency bins is denoted by M.

Amplitude Spectrum

The Amplitude Spectrum of a channel c is computed according to Equation 10. We consolidate the frequency bins into five groups of equal size and compute the average per group, resulting in five values per channel.

$$A_c = \sum_{j=1}^{M} |fft_c(j)| \tag{10}$$

Modified Median Frequency

The Modified Median Frequency determines the frequency bin j at which the amplitude spectrum A is split in two parts of equal size:

$$\sum_{j=1}^{MMDF(c)} A_c(j) = \sum_{j=MMDF(c)}^{M} A_c(j) = \frac{1}{2}\sum_{j=1}^{M} A_c(j) \tag{11}$$

Modified Mean Frequency

The average frequency in relation to the Amplitude Spectrum is computed by the Modified Mean Frequency:

$$MMNF(c) = \frac{\sum_{j=1}^{M} fft_c(j) A_c(j)}{\sum_{j=1}^{M} A_c(j)} \tag{12}$$

To avoid overfitting, we implemented a cross-validation based on the suggested procedure by Hsu et al. [5] using the radial basis function (RBF) as kernel function. The cross-validation determines the values of the kernel parameter γ and the penalty parameter C that determines to which degree false predictions on the training data are reasonable as long as the margin between the classifier and the training data is increased. We split the training data on the gesture level, i.e. we do not perform the cross-validation on single input vectors, as we want to increase the correct recognition of the complete active segment. About 10% of the training data, or at least one execution per gesture and user, is collected in a cross-validation set. Based on the remaining data, SVM models are computed for different values of C and γ, i.e. a grid search on the parameters is performed. Afterwards, we predict the label of the corresponding gesture for each execution in the cross-validation set based on each of the computed models. By comparing the result with the ground truth information, we calculate the prediction accuracy of the model. The final model is then based on the parameters yielding the highest accuracy. In our implementation, we found optimal values with $\gamma = 8$ and $C = 32$.

Classification Process

During the classification process, the same preprocessing steps as outlined above are carried out. The extracted feature vectors for each window are then given to our trained SVM which in return assigns a label for the window. To classify an active segment as a whole, we execute a majority vote on the labels of the individual windows. To avoid wrong classifications, we set a confidence threshold of 0.3, i.e. if no label is assigned to at least 30% of the considered windows, we consider the segment to not contain any of our gestures.

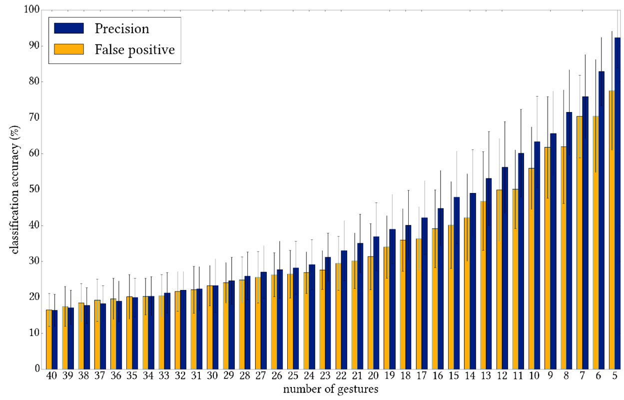


Figure 2. Classification accuracies w.r.t. different amounts of gestures and two approaches to build the set.

EVALUATION

To train our support vector machine, electromyographic data along with the name of the executed gesture, is required. We collected five labeled samples for each of our 40 gestures in randomized order from 14 users (6 females), aged from 21 to 56 (M: 37.2, sd: 10.6) all of whom had no prior experience with EMG. Three of the male participants were left-handed, as well as two female participants. With the help of our recording tool (see Figure 1), each of the participants executed 200 gestures as described before, resulting in a total of 2,800 labeled samples. As we wanted to compare our predictions with the ones made by the Myo, we insisted on the more restrictive sensor placement required by the Myo. The users were instructed to wear the armband on their dominant arm. Additionally, each user recorded one minute of data while holding his hand in a relaxed position, labeled as the *rest gesture*.

We had to delete 58 of the 2,800 executions due to incorrectly executed gestures as self-reported by the participants. We first evaluated the algorithm's ability to find active segments by inserting a gesture recording into the recording of the rest gesture at a random position, but not at the very beginning or the very end (400 ms interval). From the 2,742 evaluated samples, 91.7% could be correctly recognized, i.e. start and end points were correctly detected. When analyzing the remaining 8.3%, nearly half of them were tap gestures which did not pass the trigger threshold due to low-valued sensor readings. For the rest of the analysis, we excluded 142 gestures whose starting point was not detected. Segments without a recognizable end were further considered as they could still be classified.

We evaluated our recognition in a user-independent scenario by using a leave-one-out method and achieved an overall accuracy of 16.35% (sd: 4.54%) for our 40-gesture set, which is above chance (2.5%). The recognition results for the tap gestures are remarkably low, which could be caused by a lower number of training samples (see above) and their low peak in the raw EMG signal, which makes it difficult to discriminate them. As a system with a precision of 16% is not suitable for end-users, we tried to optimize the gesture set by following two approaches: In the first case ("false positive"), the gesture which is most often confused with other gestures is removed from the set. In the second case ("precision"), the gesture with the lowest recognition rate is removed. After decreasing the number of gestures, the classifier is again evaluated. The results of this process are depicted in Figure 2. Although we try to automatically optimize the gesture sets w.r.t. accuracy, it might be possible to achieve different results by considering other sets of gestures. As an example, we consider the set consisting of *doubletap*, *fist*, *wave_in*, *wave_out*, *three* and *ring_in* resulting in a recognition accuracy of 85.57% (sd: 8.98%) which is slightly above the accuracy of the automatically determined sets. However, evaluating all possible sets of a certain size is a very time-consuming process due to the complex classifier training, and seems not appropriate here.

To evaluate our classifier also in comparison with the state-of-the art implementation of the Myo armband, we logged all recognized gestures of the original algorithm during the data collection. We then trained our classifier with the five gestures also supported by the Myo armband (wave in/out, spread fingers, make a fist and double tap of thumb and middle finger). As before, the system was trained and tested by a leave-one-out method. In contrast, we considered all executions of the gestures, regardless of whether there was an active segment detected or not, counting not detected segments as prediction errors. Regarding the Myo, we counted an execution of a gesture as correctly recognized if the corresponding gesture was predicted correctly at least once, i.e. even if there were multiple predictions for a single execution, we counted it as correct as long as one of them was correct. While our approach achieved an overall accuracy of 95.64% (sd: 5.45%), the Myo recognized only 68.55% (sd: 13.98%) correctly. An independent t-test showed that our algorithm provided statistically significantly higher precision compared to the gesture recognition by the Myo ($t(16856) = 6,758, p < 0.001$).

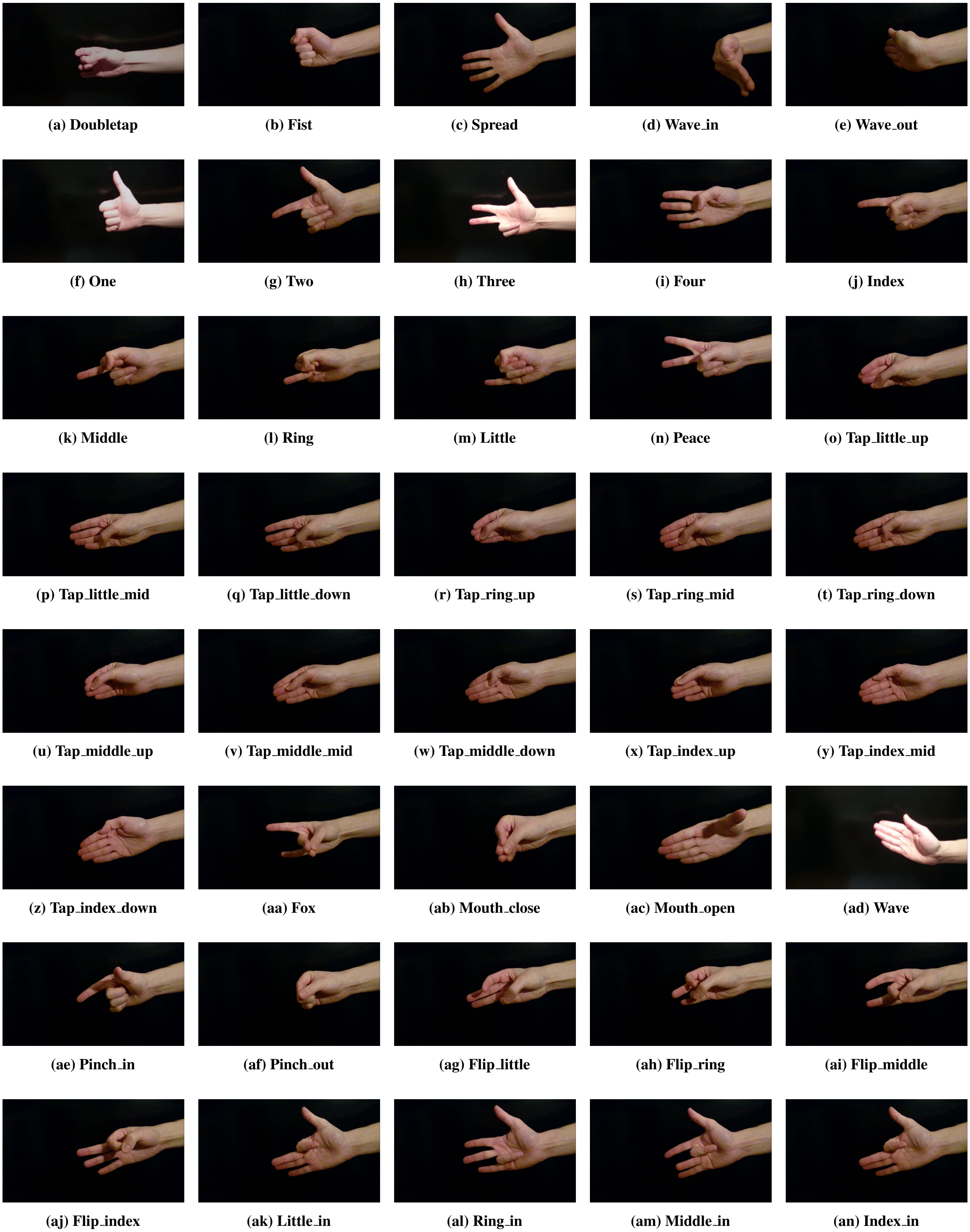
CONCLUSION AND FUTURE WORK

We presented a user-independent, real-time recognition algorithm for hand gestures based on surface EMG. Built on ten features with low computational overhead in the time as well as the frequency domain, we use a support vector machine to detect gestures in a continuous stream of sensor readings. Based on a window size of 300 ms, we are able to classify a gesture after less than half a second, and our evaluation shows that we achieve accuracies of up to 95% for gesture sets of size five. Remarkably, our algorithm significantly outperforms the current state-of-the-art recognition algorithm that the Myo armband, which we used as our sensing device, relies on.

For future work, we plan to examine whether the better recognition accuracy persists when even more participants are investigated, as it is the stated aim of the original implementation to be reliable for a wide range of users. We also plan to further investigate whether other approaches for an automatic selection of gesture sets may provide better results than the two we used. It might also be worthwhile to investigate how the results from this paper can be transferred to other EMG hardware. Additionally, it might also be promising to investigate how the recognition process can benefit from utilizing the inertial measurement unit that is integrated into the Myo wristband.

REFERENCES

1. Christoph Amma, Thomas Krings, Jonas Böer, and Tanja Schultz. 2015. Advancing Muscle-Computer Interfaces with High-Density Electromyography. In *Proc. CHI '15*. ACM, 929–938. DOI: http://dx.doi.org/10.1145/2702123.2702501
2. Marcus Georgi, Christoph Amma, and Tanja Schultz. 2015. Recognizing Hand and Finger Gestures with IMU Based Motion and EMG Based Muscle Activity Sensing. In *Proc. BIOSTEC '15*. 99–108. DOI: http://dx.doi.org/10.5220/0005276900990108
3. Fredric J. Harris. 1978. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proc. IEEE* 66, 1 (1978), 51–83.
4. Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body as an Input Surface. In *Proc. CHI '10*. ACM, 453–462. DOI: http://dx.doi.org/10.1145/1753326.1753394
5. Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, and others. 2003. A Practical Guide to Support Vector Classification. (2003).
6. Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proc. UIST '16*. ACM, 321–333. DOI: http://dx.doi.org/10.1145/2984511.2984582
7. Yun Li, Xiang Chen, Jianxun Tian, Xu Zhang, Kongqiao Wang, and Jihai Yang. 2010. Automatic Recognition of Sign Language Subwords Based on Portable Accelerometer and EMG Sensors. In *Proc. ICMI-MLMI '10*. ACM, Article 17, 7 pages. DOI: http://dx.doi.org/10.1145/1891903.1891926
8. Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. 2016. EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. In *Proc. CHI '16*. ACM, 2332–2342. DOI: http://dx.doi.org/10.1145/2858036.2858093
9. Ganesh R. Naik, Dinesh Kant Kumar, Vijay Pal Singh, and Marimuthu Palaniswami. 2006. Hand Gestures for HCI Using ICA of EMG. In *Proc. HCSNet Workshop VisHCI '06*. Australian Computer Society, Inc., 67–72. http://dl.acm.org/citation.cfm?id=1273385.1273397
10. Angkoon Phinyomark, Chusak Limsakul, and Pornchai Phukpattaranont. 2009. A Novel Feature Extraction for Robust EMG Pattern Recognition. *Journal of Computing* 1, 1 (2009), 71–80.
11. Siddharth S. Rautaray and Anupam Agrawal. 2015. Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey. *Artificial Intelligence Review* 43, 1 (2015), 1–54. DOI: http://dx.doi.org/10.1007/s10462-012-9356-9
12. Ali-Akbar Samadani and Dana Kulic. 2014. Hand Gesture Recognition based on Surface Electromyography. In *Proc. IEEE Engineering in Medicine and Biology Society '14*. 4196–4199. DOI: http://dx.doi.org/10.1109/EMBC.2014.6944549
13. T. Scott Saponas, Desney S. Tan, Dan Morris, and Ravin Balakrishnan. 2008. Demonstrating the Feasibility of Using Forearm Electromyography for Muscle-Computer Interfaces. In *Proc. CHI '08*. ACM, 515–524. DOI: http://dx.doi.org/10.1145/1357054.1357138
14. T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-Available Input with Muscle-Computer Interfaces. In *Proc. UIST '09*. ACM, 167–176. DOI: http://dx.doi.org/10.1145/1622176.1622208
15. T. Scott Saponas, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. 2010. Making Muscle-Computer Interfaces More Practical. In *Proc. CHI '10*. ACM, 851–854. DOI: http://dx.doi.org/10.1145/1753326.1753451
16. Eric Scheme and Kevin Englehart. 2014. On the Robustness of EMG Features for Pattern Recognition Based Myoelectric Control; A Multi-Dataset Comparison. In *Proc. IEEE Engineering in Medicine and Biology Society '14*. 650–653. DOI: http://dx.doi.org/10.1109/EMBC.2014.6943675
17. Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. *Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs*. Springer, 267–280. DOI: http://dx.doi.org/10.1007/11555261_24
18. Robert Y. Wang and Jovan Popović. 2009. Real-time Hand-tracking with a Color Glove. *ACM Trans. Graph.* 28, 3, Article 63 (July 2009), 8 pages. DOI: http://dx.doi.org/10.1145/1531326.1531369
19. Mark Weiser and John Seely Brown. 1997. Beyond Calculation. Copernicus, Chapter The Coming Age of Calm Technolgy, 75–85.
20. John Weissmann and Ralf Salomon. 1999. Gesture Recognition for Virtual Reality Applications Using Data Gloves and Neural Networks. In *Proc. IJCNN '99*, Vol. 3. 2043–2046 vol.3. DOI: http://dx.doi.org/10.1109/IJCNN.1999.832699
21. Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. 2011. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41, 6 (Nov 2011), 1064–1076. DOI: http://dx.doi.org/10.1109/TSMCA.2011.2116004
22. Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. 2009. Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors. In *Proc. IUI '09*. ACM, 401–406. DOI: http://dx.doi.org/10.1145/1502650.1502708
23. Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proc. UIST '15*. ACM, 167–173. DOI:http://dx.doi.org/10.1145/2807442.2807480



(a) Doubletap (b) Fist (c) Spread (d) Wave_in (e) Wave_out

(f) One (g) Two (h) Three (i) Four (j) Index

(k) Middle (l) Ring (m) Little (n) Peace (o) Tap_little_up

(p) Tap_little_mid (q) Tap_little_down (r) Tap_ring_up (s) Tap_ring_mid (t) Tap_ring_down

(u) Tap_middle_up (v) Tap_middle_mid (w) Tap_middle_down (x) Tap_index_up (y) Tap_index_mid

(z) Tap_index_down (aa) Fox (ab) Mouth_close (ac) Mouth_open (ad) Wave

(ae) Pinch_in (af) Pinch_out (ag) Flip_little (ah) Flip_ring (ai) Flip_middle

(aj) Flip_index (ak) Little_in (al) Ring_in (am) Middle_in (an) Index_in

Figure 3. The 40 gestures used in our evaluation. Images are taken from the instruction videos shown to the participants in our recording environment.