Gesture Recognition for Navigation in Multi-Dimensional Data

Mariagrazia Fugini
Dipartimento di Elettronica
Informazione e Bioingegneria
Politecnico di Milano
Milano, Italy
mariagrazia.fugini@polimi.it

Jacopo Finocchi
Dipartimento di Elettronica
Informazione e Bioingegneria
Politecnico di Milano
Milano, Italy
jacopo.finocchi@mail.polimi.it

Giulia Trasa
Dipartimento di Elettronica
Informazione e Bioingegneria
Politecnico di Milano
Milano, Italy
giulia.trasa@mail.polimi.it

***Abstract*— This paper presents an environment where a set of data collected by IoT sensors can be navigated through gesture recognition algorithms and through virtual reality tools. In particular, the aim of recognizing human gestures is to give commands through a wearable control device. The paper presents the Seamless project, where the recognition and visualization methods have been deployed, and concentrates on the problem of automatically understanding a set of gestures using Machine Learning. The employed model and the results are illustrated.**

Keywords—Gesture Recognition, Wearable Control Device Supervised Machine Learning, Dynamic Time Warping, Inertial Measurement Unit.

I. Introduction

This paper deals with the research activity on IoT-captured signals from which movements need to be extracted, being movements of structures, people, environmental objects, and so on. The research has been performed as part of the "*Seamless*" Project, conducted in a cooperation among Politecnico di Milano - DEIB, Next Industries[1] and Digisoft System Engineering (DSE)[2]. The goal of *Seamless*, funded by Regione Lombardia – Italy, is the development of software and algorithms dedicated to analyze movements of structures (bridges, viaducts, buildings) due to both inherent moves and territorial events (earthquakes, vehicle solicitations, etc.). Movements are detected by analyzing data collected from sensors positioned on the monitored structure and collected via an Internet of Things (IoT) platform.

We investigated a sub problem of movements, namely the *recognition of human gestures* obtained by analyzing inertial sensor data collected by wearable devices. This problem, which is the object of this paper, was tackled due to the availability of the Tactigon-Skin™ device (a wearable similar to a glove endowed with sensors) produced by Next Industries and equipped with an Inertial Measurement Unit (IMU). In the Project, we tested a solution to recognize hand gestures performed while wearing Tactigon-Skin (a small hand-held device), to enable its use as a man-machine control device, aimed at the navigation of three-dimensional visual representations of structural data.

In particular, we carried out *gesture analysis* algorithms:

1. Based on accelerometer and gyroscope data only, with no additional equipment (cameras, etc.);
2. In real time;
3. In a user-independent manner;
4. With no gesture start or end markers;
5. Aimed at supporting cooperative navigation in a multi-dimensional space of data, with Virtual Reality potentialities.

In addition to recognizing isolated gestures, our approach recognizes *consecutive hand movements* without providing any special signal (e.g., clicking a button) to identify the start/ end of each gesture, in order to achieve both a better user experience and a deeper testing of the proposed algorithms.

Another goal of gesture recognition in the Project is to perform *a continuous data flow analysis* in *real time*, as the response speed of the recognition software must allow a prompt man-machine interaction and possible interventions for risk mitigation [1].

Moreover, the developed solution had to be independent of the specific user wearing the device, so as not to require a training activity of the system for each user.

In *Seamless*, DSE developed Virtual Reality and Augmented Reality tools [2] able to: i) display the environment of IoT multi-dimensional data; ii) use the gestures recognition algorithm outputs to control the navigation. In particular, the system developed by DSE provides a *virtual environment* where multiple subjects can interact collaboratively by connecting remotely through the web to a cooperative space of IoT "objects". This environment can be set up quickly in any room, even with no pre-setting or knowledge of the physical/virtual layout, by using a simple and mobile equipment consisting of a set of IoT devices. Only a quick recognition of the geometry of each room, where remotely connected users are located, is required to avoid collisions with real obstacles. The purpose is to provide a cooperative solution to navigation in multi-dimensional data, to be applied in business, industrial, and scientific or medical frameworks. This enables, for example,

[1] https://www.nextind.eu
[2] https://www.gruppodse.it

a virtual meeting where data can be moved around to examine structural elements displayed in 3D, or to examine a 3D model of a monitored structure (e.g., a mouth by the dentist for healthcare or a bridge structure monitored by civil engineers).

As a display device, a headset with a Virtual Reality viewer has been developed by DSE, equipped with a microphone and with headphones. The purpose is to allow voice and video communication. The headset is controlled by a software application based on the Unreal Engine, which is interfaced with the Tactigon-Skin provided by Next Industries, as a gesture control input device.

The *Seamless* environment has been used to test a set of algorithms for gesture recognition based on Machine Learning (ML). In the first phase of the project, that is described in this paper, we experimented the *Dynamic Time Warping* (*DTW*) algorithm [3], particularly suitable for the analysis of time series and used in various fields of application, from Speech Recognition to the recognition of human activities.

The paper is organized as follows. In Section 2, we review the literature in the field. In Section 3, we illustrate our solution to gesture recognition. In Section 4, results are discussed.

II. Related Work

Gestures recognition provides an automated yet natural way for man–machine interaction. Many works for hand gesture recognition have been published in the last years, based on different technologies, illustrated in the following considering those that are most related to our approach.

The majority of related work solutions use additional sensor technology, combining inertial data with visual information from a camera or performing a data fusion between inertial data and another data source, like infrared or magnetic sensors. A vision-based approach is proposed in [3] to build a dynamic hand gesture recognition system. An infrared light-based method is proposed in [4].

In [5] gestures capturing and recognition is based on inertial and magnetic measurements units (IMMUs) assembled on a fully cabled glove. The paper describes recognition methods based on Extreme Learning Machine, a kind of feed-forward neural networks, both for static and dynamic gestures classification.

A method based on mobile device sensors is proposed in [6]. A new statistical quantization approach is introduced, using gesture-specific codebooks, Hidden Markov Models, and error models for gesture sequences. During classification, it exploits the error model to explore multiple feasible HMM state sequences. In [7] an accelerometer-based pen-type sensing device is presented, with a user-independent hand gesture recognition solution. An effective segmentation algorithm is developed to identify individual basic gesture motion intervals, where each complex gesture is segmented into several basic gestures. Based on the kinematics characteristics of the basic gestures, features are extracted to train a feedforward neural network model. For recognition of basic gestures, input gestures are classified directly by the neural network classifier, while complex gestures go through an additional similarity matching procedure, to identify the most similar sequences.

For the analysis of inertial data, several researchers apply the DTW algorithm [3, a technique that evaluates the distance between two sequences through a non-linear distortion with respect to an independent variable, such as time. Hence, it is suitable for the recognition of time series of data. A Kinect sensor is used in [8] to collect depth data in order to track and recognize hand gestures in real time. The k-curvature algorithm is employed to locate the fingertips over the contour, and DTW is used to select gesture candidates and to recognize gestures by comparing an observed gesture against a series of pre-recorded reference gestures. In [9] RGB-D sensors (a combination of RGB image with depth information) are used, giving an easy way to acquire 3D points and track them using the depth information, whereas the DTW algorithm is used as a classification method. The proposed method is an approach focused exclusively on the use of geometric information. An algorithm detects and tracks the 3D hands movement, then it smooths and simplifies the gesture to remove noise and capture the key points. The last step uses DTW as a classification method to recognize gestures. A nearest-neighborhood algorithm is combined with DTW to find the closest gesture, wrt a given cost distance function. In [10] a classification technique is proposed based on a DTW adaptable algorithm. The utility of DTW-Dependent and DTW-Independent varies on an instance-by-instance basis, and a DTW adaptive approach predicts at run time which of them is more likely to be correct.

A continuous hand gestures recognition technique is presented in [11], that is capable of continuous recognition of hand gestures using a three-axis accelerometer and gyroscope embedded in a smart device. The influence of hand unstableness is managed through a gesture-coding algorithm, which also performs data compression. An automatic gesture-spotting algorithm detects the start and end-points of meaningful gesture segments. Similarly, a gesture is recognized by comparing the gesture code against a gesture database using a DTW algorithm.

As described above, the gesture recognition tasks are usually based on ML techniques to manage the high variability in the input data.

III. The Developed Solution

In this section, we present the approach to gesture recognition by presenting the collected data and their analysis.

A. Sensor Data

As a gesture set for the experimentation, we defined a series of simple translations or rotations and a couple of more complex two-dimensional gestures. The set of considered motions comprises the gestures reported in Table 1.

TABLE 1 - THE DEFINED GESTURE SET

Gesture Type	Gesture Name
Simple Translation	Forward
	Backward
	Up
	Down
	Right
	Left
Simple Rotation	Pitch Right
	Pitch Left
	Roll Up
	Roll Down
	Yaw and return
Complex Gesture	Circle
	Square

Since the aim of our algorithm is to support man-machine interaction, the chosen gestures are suitable for intuitive command input, and are also designed to possibly generate new commands by concatenation of single inputs.

In using Tactigon-Skin to execute gestures, the user has to respect some basic rules, namely:

- performing neither too fast nor too slow movements, taking not less than 0.5 seconds and no more than 1.5 seconds for each gesture;
- pause the hand movement for half a second at least, before starting each next gesture.

These constraints were considered to be compatible for a practical use of the device as a gesture command interface.

A 6-axis IMU chip, based on Micro Electro-Mechanical Systems (MEMS) technology, collects inertial data, measuring three-axis acceleration and three-angular velocities with 50Hz frequency.

From a first analysis of the data samples, several alteration factors emerged about the acceleration data, namely, data show:

- *noise*: a widespread high frequency noise gives sensor data a jagged course;
- *gravity*: its presence translates vertical acceleration by a fixed component of about -9.81 m/s2;
- *drift*: when considering long traces, we often observe in acceleration data the appearance of small offsets, or they do not reset to zero when the device stops, due to loss of calibration or to measurement errors.

In order to obtain the velocities and trajectories of the hand movement by integrating the available data, data should be filtered to remove the noise and some fixed components.

This first processing step can be accomplished by applying simple low-pass and high-pass filters, but apart from the removal of gravity acceleration, filtering was not found to be critical for our goals. In fact, the effects of noise are mitigated by the data normalization step performed after the gesture segmentation, while the drift effects are reduced by the short duration of a single gesture and by leveling the acceleration values to zero at the ending of each gesture.

B. Gesture Segmentation

To achieve gesture recognition in *quasi real-time* from an inertial data flow analysis, the next task to be completed was the development of a *segmentation algorithm* capable of extracting the subset of data corresponding to a single gesture from a continuous data flow. The released version of the segmentation algorithm is able to detect the short pause that separates consecutive gestures, based on angular velocities, linear accelerations and a moving average of the acceleration gradients. These values are evaluated against a set of thresholds, which were obtained experimentally. A higher weight is assigned to the angular velocities, since they so result less affected by drifts. When a candidate gesture exceeding the shortest allowed gesture duration, is followed by a pause condition that in turn exceeds the minimum allowed pause duration, the gesture is detected. A finer-grained analysis is then performed at both ends of the data segment, to recover any possible values below threshold at the start point and at the concluding point of the gesture.

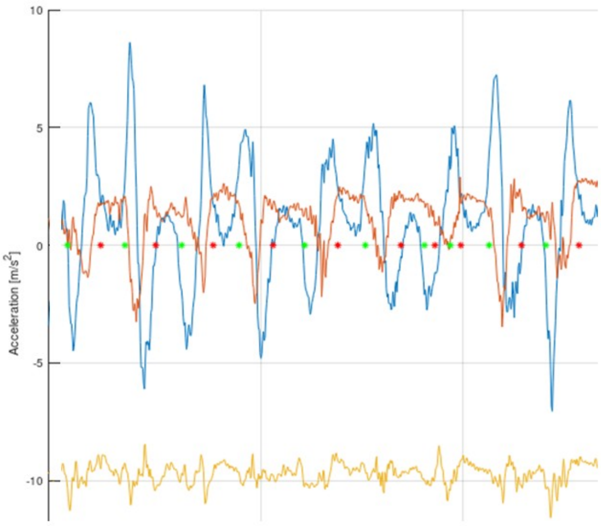

Fig. 1. Segmentation of Left and Right gesture sequence

An example of successful segmentation, despite the data disturbances is shown in Figure 1, where the green dots mark any gesture start and the red dots mark any gesture conclusion.

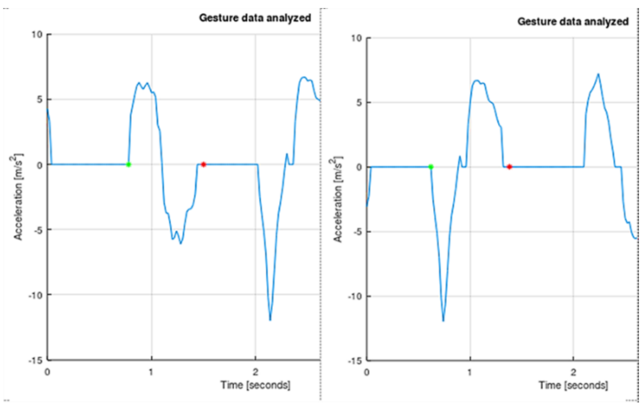

Fig. 2. Two segmented guestures: Down (left) and Up (right)

C. Gesture Normalization

After the detection of a gesture data segment, the corresponding acceleration path is normalized in length and in amplitude. The length normalization performs a reduction of the samples size to a standard number of values. In the amplitude normalization, the acceleration peak value is standardized to the value of 100% and all other values are proportionally computed. The order of the two operations is important to keep the information value of gesture data.

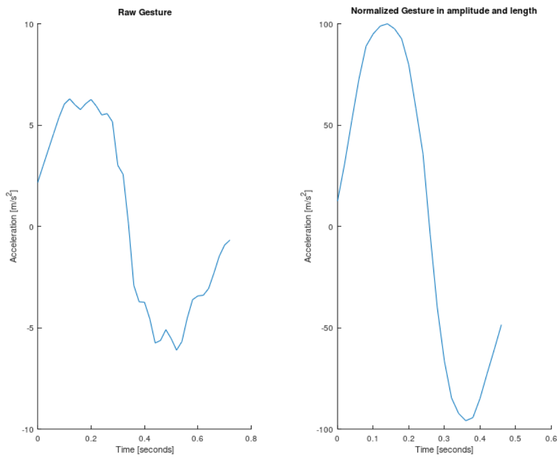

Fig. 3. Raw gesture data and normalized data

Figure 3 shows an example of gesture data normalization. These processing steps allow bringing together data segments corresponding to short and long span gestures as well as faster and slower gestures, that otherwise show very different raw acceleration values.

D. Gesture Classification

The gesture recognition task was carried out as a problem of data classification. The central problem of recognizing the performed gesture is traced back to a problem of classification of the input data set.

In an initial phase, therefore, an overview was made of the possible classification algorithms to be used in the project, summarized in Figure 4.

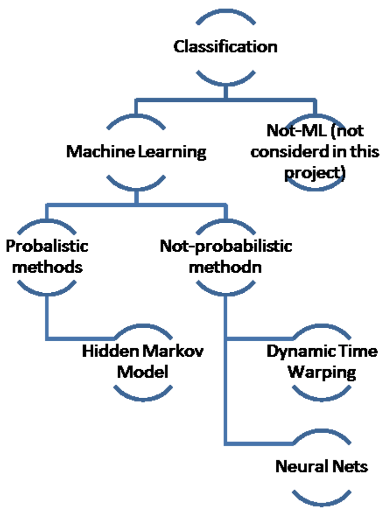

Fig. 4. Available methods for data classification.

Among the many ML techniques available for a data classification task, we first experimented the DTW algorithm.

Our approach was to privilege the possibility of arriving at a practical usable result during the Project, developing a functional prototype, rather than seeking the optimal approach by experimenting with a wider spectrum of options. Instead of a systematic comparison between the different techniques available, we therefore concentrated on the DTW method, which from the literature on the subject, is considered suitable for the recognition of "time series", that is, of time sequences of values. This technique allows for the alignment between two sequences, and leads to the measurement of a distance between the two aligned sequences. DTW algorithm is particularly useful for sequences where individual components have characteristics that vary over time, and for which the simple linear expansion or compression of the two sequences does not yield satisfactory results.

An advantage of DTW is that a small number of sample data sequences are sufficient for the training phase. Moreover, it does not require the evaluation of statistical features characteristic of each gesture, but can be applied directly to the acceleration and angular velocity data provided by the IMU. Sensor data, recorded during specific training sessions, were manually selected and classified, to form a set of *reference gesture models*, after an automatic gesture segmentation. The approach is therefore a *supervised ML solution*.

To simplify the classification problem, we took advantage of the consideration that the accelerations or angular velocities of our gesture set were supposed to affect mostly a single spatial dimension at a time. In order to reduce the number of comparisons between the detected gesture and the gesture models, a preliminary classification stage was introduced. In this coarse evaluation, it is assessed whether the gesture is a translation or a rotation. A feature combining the maximum inclination reached by the device and the maximum angular velocity is evaluated against a threshold, to discriminate the rotation gestures from others. Then, the main axis is determined along which the movement occurred, to reduce the set of gesture models to be compared.

Finally, the DTW algorithm is used to compute the distance between the normalized gesture data and each gesture model. Where the shortest distance is found, the corresponding classification is assigned and the gesture is recognized. Since a minimum distance is always found, if the distance exceeds a given maximum value, the data are classified as an *uncertain gesture*.

In a second moment, the two-dimensional gestures (circle and square) were introduced and some adjustments were applied to the classification task. The thresholds based solution in fact was able to discriminate rotations from translations but it resulted inaccurate to detect more complex gestures. For this purpose we successfully experimented a single-layer feed-forward neural network, which is given the peak values for the six inertial data dimensions and returns a preliminary classification of the gesture in the three macro-categories corresponding to one-dimensional translation, one-dimensional rotation or two-dimensional gesture.

IV. THE SOFTWARE PROTOTYPE

A software prototype has been developed for gesture recognition of gestures performed using the T-skin wearable control device.
The different steps of the inertial data analysis process described so far are the basis of the gesture recognition prototype developed during the project, as summarized in Figure 5.

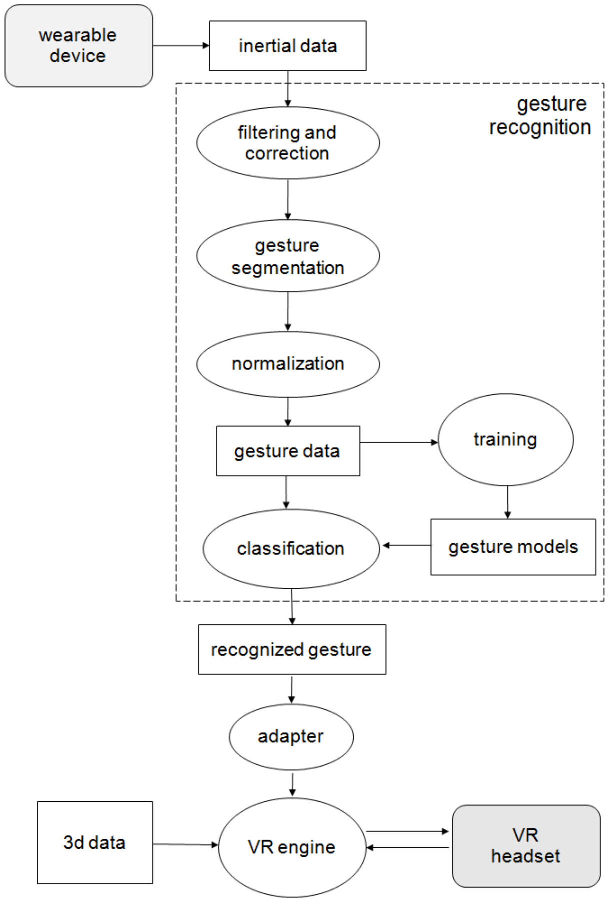

Fig. 5. The gesture recognition process in the Seamless environment.

The selected development environment is Octave, an open source version of "Matlab, a flexible tool for our development of algorithms, postponing subsequent developments in C or Python to a later stage.

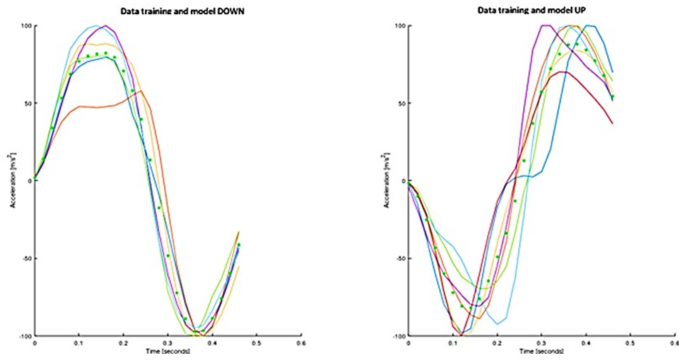

Fig. 6. Training gestures and models for Down and Up gestures

V. SUMMARY OF RESULTS

During the experimentation, 32 data registration sessions were carried out, involving 5 different users, in order to verify the user independence of the developed solution. Users were shortly instructed about the type of gestures they could perform.

Each test recording included a sequence of different gestures, from 10 to 40 for each session, for a total number of 652 gestures. In some sessions, they followed a fixed sequence of gestures while in others they were free to choose the gestures on the fly.

The recognition software analysed each sequence separately and in all correctly recognized 587 gestures, equal to 90.03%, while 65 were not recognized or incorrectly classified.

Two types of *errors* emerged during the tests: classification errors and segmentation errors.

- Classification errors are attributable to the classification module, which, starting from the correct section of data, produces an incorrect classification output. Some of the main causes of classification errors are due to noise in the data, to the intensity with which a gesture is made, or to the incorrect position of the wearable device on the hand.

- Segmentation errors are due to the segmentation module, which delivers an incorrect portion of data to the classification module. Segmentation errors occur in the case of partial data corresponding to a fraction of the gesture or on the contrary from the union of a gesture with the following gesture. This second type of error is mainly due to the way gestures are made, when two consecutive gestures are separated by a too long or too short pause.

In particular, the errors were determined by an incorrect segmentation in 38.7% of cases and by an incorrect classification in the remaining 61.3%.

The results can be summarized as in Figure 7.

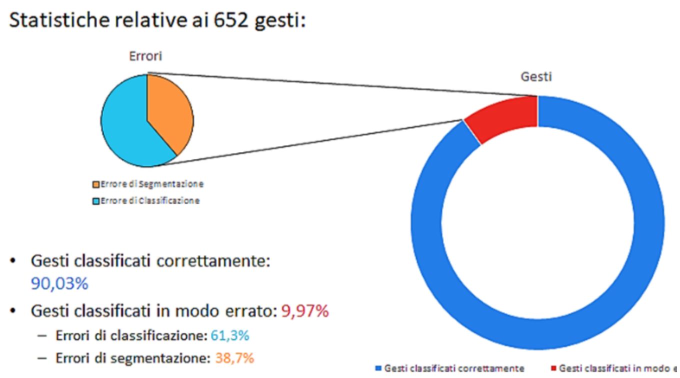

Fig. 7. Final test sessions results with DTW approach

VI. CONCLUDING REMARKS

The paper has presented practical experimentations about gesture recognition performed through Data Warping Algorithms to train a wearable device. In particular, we have illustrated the classification steps and the obtained results.

Among the future developments of the classification module, we are considering a possible transition to the three-dimensional version of DTW algorithm, in particular for gestures involving multiple axes.

Furthermore, we are starting to test different ML algorithms, such as Neural Networks and Hidden Markov Models. Neural networks, which we applied for the gesture macro-category classification task, will be tested to directly classify each specific gesture as an alternative to employing gesture models.

REFERENCES

[1] Teimourikia, M., Fugini, M. and Raibulet, C., 2017, June. Run-Time Security and Safety Management in Adaptive Smart Work Environments. In 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 256-261). IEEE.

[2] Peters, E., Heijligers, B., de Kievith, J., Razafindrakoto, X., van Oosterhout, R., Santos, C., Mayer, I. and Louwerse, M., 2016, September. Design for collaboration in mixed reality: Technical challenges and solutions. In 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES) (pp. 1-7). IEEE.

[3] BERNDT, D.J.; CLIFFORD, J. Using dynamic time warping to find patterns in time series. In KDD workshop 1994 Jul 31 (Vol. 10, No. 16, pp. 359-370).

[4] SINGHA, Joyeeta; ROY, Amarjit; LASKAR, Rabul Hussain. Dynamic hand gesture recognition using vision-based approach for human–computer interaction. Neural Computing and Applications, 2018, 29.4: 1129-1141.

[5] NISHIHARA, H. Keith, et al. Hand-gesture recognition method. 9,696,808, 2017.

[6] FANG, Bin, et al. 3D human gesture capturing and recognition by the IMMU-based data glove. Neurocomputing, 2018, 277: 198-207.

[7] TOOTAGHAJ, Diman Zad, et al. High Five: Improving Gesture Recognition by Embracing Uncertainty. arXiv preprint, arXiv:1710.09441, 2017.

[8] XIE, Renqiang; CAO, Juncheng. Accelerometer-based hand gesture recognition by neural network and similarity matching. IEEE Sensors Journal, 2016, 16.11: 4537-4545.

[9] PLOUFFE, Guillaume; CRETU, Ana-Maria. Static and dynamic hand gesture recognition in depth data using dynamic time warping. IEEE transactions on instrumentation and measurement, 2015, 65.2: 305-316.

[10] SOARES, Alan dos Santos; APOLINÁRIO JR, Antonio L. Real-time 3D gesture recognition using dynamic time warping and simplification methods. 2017.

[11] SHOKOOHI-YEKTA, Mohammad; WANG, Jun; KEOGH, Eamonn. On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In: Proceedings of the 2015 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2015. p. 289-297.